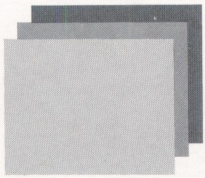
Genesis



SOFTWARE SOLUTIONS

GENESIS

User Guide

Copyright © Software Solutions 1990

Neither the whole nor any part of the information contained in this Guide, nor the software contained on the disc supplied with it may be adapted or reproduced in any material form except with prior written approval of Software Solutions.

First Edition - January 1990

Published by Software Solutions

All trademarks acknowledged.

Your attention is drawn to the End-User License Conditions at the end of this User Guide.

This User Guide was produced using the *Acornsoft VIEW* word processor and *Acorn Desktop Publisher*. Final copy was produced using a Gestetner Lasers PostScript printer.

Genesis is written in BBC Basic 5 and the final version compiled using the DABS Press *Archimedes Basic Compiler*.

Genesis was designed in conjunction with Hampshire Local Education Authority. Many thanks to Martyn Wilson and his team of Advisory Teachers.

Thanks also to Cleveland LEA and Waltham Forest LEA for their help during the trialling of the software.

Software Soltutions can be contacted at:

Broadway House
149-151 St Neots Road
Hardwick, Cambridge CB3 7QJ
0954 211760

Euclid and Mogul can be obtained from:

Ace Computing
27 Victoria Rd
Cambridge CB4 3BW
0223 322559

CONTENTS

1	Introduction	1
2	Installation	3
3	Creating an Application	7
4	Running an Existing Application	17
5	Further Types of Data	23
6	Databases	31
7	Browsing	39
8	The Complete Picture	43
9	The Genesis Filer	55
10	Script Files	59
11	Genesis Script Language	69
Appendix A:	Customising your Application	85
Appendix B:	Cross Application Linking	87
Appendix C:	Structure of a Genesis Application	88
Appendix D:	Index Files	89
Appendix E:	Internationalisation	90
Index		91

1 INTRODUCTION

Genesis is an extremely powerful, yet simple to use system which allows you to create your own RISC OS applications. These applications consist of a collection of linked pages of information.

A typical page contains several rectangular areas called frames. Each of these frames may have information attached to it. For example, a frame may display text, graphics or even an animated sequence. Alternatively it may contain an icon which performs an action such as running a separate program or playing a tune.

Movement from one page to another is achieved via links from these frames. There can be many routes leading from a particular page because each frame is allowed to be linked to a different destination. This means that information can be presented more naturally than is possible with traditional database systems.

The layout of a page can be altered entirely through use of the mouse. Frames can be moved and resized and their contents are automatically updated; graphics are scaled to fit and text is formatted appropriately. Further control is provided by means of menus obtained from individual frames and pages. For example, the menu from a text frame provides options for changing the text style, size and colour.

Uses for Genesis

Genesis is ideal for project work carried out by a group. It allows individuals or small teams to concentrate on a particular area which interests them and create their own page containing the information they have gathered. When all pages are complete, they can then be linked together in an appropriate order to form the final application. This application can then be used as a resource for others to access and individual pages can be printed out so that copies of the information can be kept alongside written work.

Alternatively Genesis may be used as a database creator. By defining a page that contains the fields normally associated with a database record and making copies of this page, a database of several records can quickly be built up. The advantage of Genesis is that it doesn't restrict you to a particular format. If necessary, extra fields can be added to a particular record or existing ones extended. In addition, records can contain digitized pictures, diagrams or pieces of music as well as the traditional textual information. Once a database has been created, the powerful searching facilites built into Genesis can be used to locate information.

The example applications provided with Genesis show some of the many different ways in which it can be applied. However, the list of possible uses is endless. Using Genesis you can quickly and easily create an application which will present any collection of information in an attractive and flexible manner.

2 INSTALLATION

Your Genesis system consists of two discs. The first is the Genesis Applications Editor disc and it contains the following:

!Genesis The Genesis Editor which is used for creating and altering applications.

!Browser The Genesis Browser which allows existing applications to be accessed but not altered.

!GenFiler The Genesis Filer which is used for displaying the resources and pages associated with an application.

!GenLib A library of resources such as drawings, sprites etc which can be used in your own applications.

The second disc is the Genesis Example Applications disc which contains two example Genesis applications:

!AboutGen An example which provides information about Genesis itself.

!Europe A geographical example providing information about some countries and cities of Europe.

Genesis is designed for use under the RISC OS desktop. On the distribution discs, the software is present in a ready to run form. However, it is strongly recommended that you should install the system onto either another pair of floppy discs or a hard disc. You should then put the original distribution discs away in a safe place.

If you are installing onto floppy discs, you must have two blank, formatted discs ready to install the software onto. These will become your Genesis work disc and example disc. Users with only a single drive should read the next section as well as this one before starting the installation in order to avoid unnecessary disc swapping.

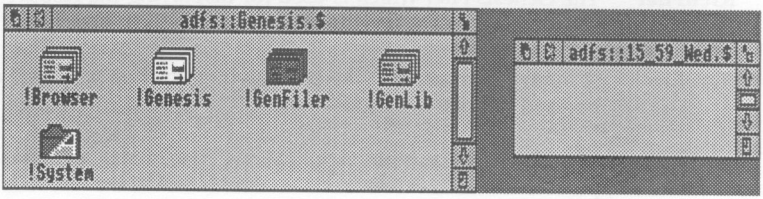
To install Genesis, first enter the RISC OS Desktop. It may be that your Archimedes automatically starts up in the Desktop in which case you have nothing to do. However, if this is not the case, you can enter it from BASIC or from the supervisor prompt by typing:

```
*Desktop
```

Place the Genesis Applications Editor disc in drive 0 and, using the mouse, click on the icon representing this drive. A directory viewer will appear showing the contents of the disc.

Using the mouse, open a directory viewer onto the target disc. If this is your hard disc, click on the hard disc icon on the icon bar. If it is a floppy disc, insert the floppy disc in any drive and click on the appropriate drive icon.

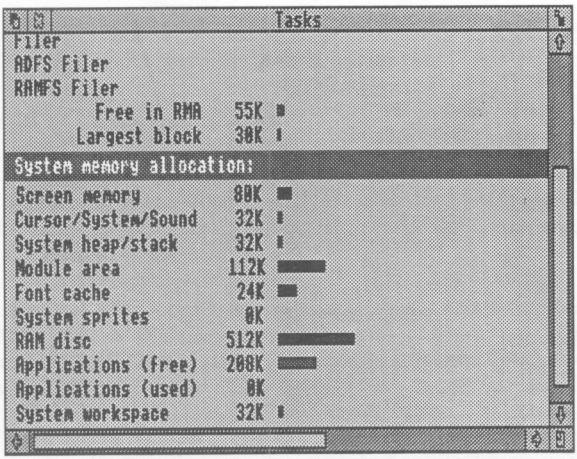
You should now have two directory viewers on the screen. One representing the Genesis distribution work disc and the other representing your target work disc as shown below:



Now, using the mouse, drag each of the icons from the Genesis distribution directory to the target directory. When this has been completed, repeat the process for the second disc.

Note for users with only a single drive

The copying of the directories from one floppy to another on a single drive machine will require a very large number of disc swaps. This can be avoided by using the RAM filing system. You should first set up the RAM disc to be large enough to hold all the files. This can be done from the Desktop using the task manager window. Click the menu (middle) button on the Archimedes 'A' symbol on the icon bar at the bottom of the screen. Then point at and select the **Task display** option from the menu which pops up. The Tasks window will appear. Using the scroll bar at the right side of the window, scroll the contents until you can see the entry for the RAM disc. Then, using the mouse, drag the red bar adjacent to this entry to the right to increase the size of the RAM disc. 512K bytes will be required for the copy process:



Now open a directory viewer onto the RAM disc and perform the dragging operations described above to copy all of the icons from the distribution disc to the RAM disc. Finally, insert the target disc and drag the directories to the target directory.

If you don't have 512K available, you can, instead, use just 128Kbytes which is sufficient to allow the copying to be performed in stages, one directory at a time.

When the copying process has been completed, you should tidy up by deleting the contents of the RAM disc and resetting its size to the original value. To delete the contents point at the RAM disc directory, click the menu button and select the **Select all** option. Display the menu again, and this time slide off the **Selection** option to obtain a second menu and from this select the **Delete** option. This will delete all the contents of the RAM disc. The RAM disc size can then be reset using the task manager as described above.

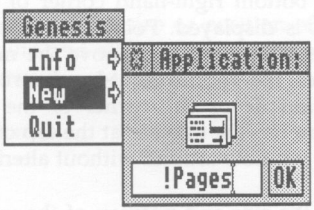
3 CREATING AN APPLICATION

The first thing you should try doing with Genesis is to create an application of your own. This will show you how simple it is to use.

Making a new blank application

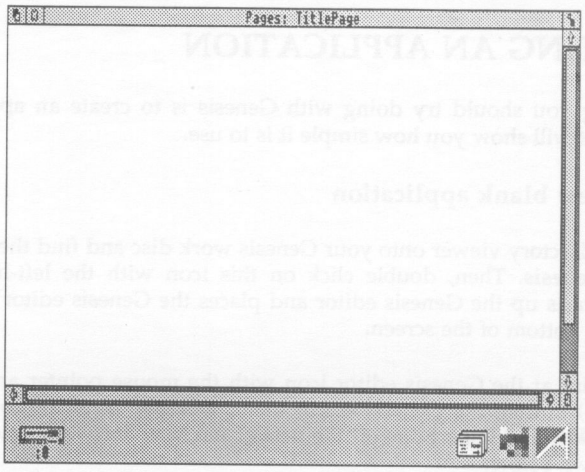
First, open a directory viewer onto your Genesis work disc and find the icon which is labelled **!Genesis.** Then, double click on this icon with the left-hand mouse button. This starts up the Genesis editor and places the Genesis editor icon on the icon bar at the bottom of the screen.

When you point at the Genesis editor icon with the mouse pointer and press the menu (middle) button, the Genesis editor menu will pop up. This contains three entries: **Info, New** and **Quit.** Slide off the side of the **New** entry and a window will appear as shown below:



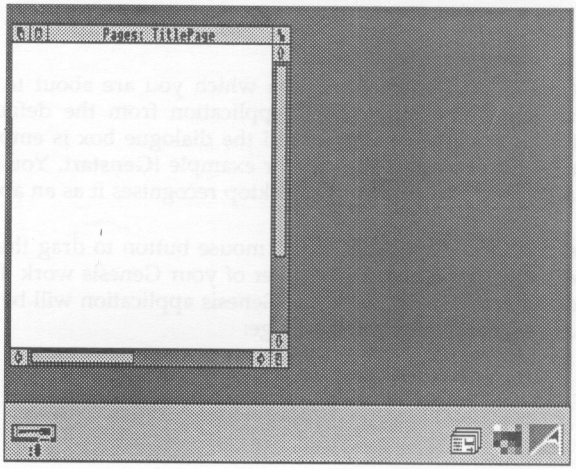
This represents the new Genesis application which you are about to create. You can, if you wish, alter the name of this application from the default which is **!Pages.** To do this, press the delete key until the dialogue box is empty and then type in the name which you want to use, for example **!Genstart.** You should start the name with a '!' character so that the desktop recognises it as an application.

When you have done this, use the left-hand mouse button to drag the application icon from the window to the directory viewer of your Genesis work disc. As soon as you drop the icon into the directory, the Genesis application will be created and a window will appear containing the first page:



Initially, this window covers the whole screen except for the icon bar. You can use the adjust size icon in the bottom right-hand corner of the window to alter the amount of the page which is displayed. Point at this icon, press the left mouse button and with the button still held down, move the mouse pointer around the screen. As the pointer moves, the bottom right-hand corner of the window moves with it. Releasing the button drops the corner of the window at the current position. In addition, you can use the title bar at the top of the window in a similar manner, to move the whole window around without altering its size.

Try these out now, to alter the size and position of the window so that the screen looks similar to that below:



Creating frames

The next thing to do is to create a frame within this page. This is achieved using the following sequence:

1. Point at where you want the top left-hand corner of the frame to be

2. Press and hold down the *left* mouse button

3. Move the mouse pointer to the bottom right-hand corner of the frame

4. Release the mouse button

During step 3, a dotted rectangle will appear indicating the current extent of the frame. The top left-hand corner of this rectangle is fixed to the position you chose in step 1, but the bottom right follows the mouse pointer as you move it around the screen. When you finally release the mouse button, a black rectangle will be drawn marking the border of the frame which you have just created.

It doesn't matter if you don't get the size or position of the frame right first time, you can move frames around and alter their shape as often you like. Both actions are performed in a similar way.

To alter the position of a frame:

1. Point inside the frame

2. Press and hold down the *left* mouse button

3. Move the mouse pointer to reposition the frame

4. Release the mouse button

As the mouse pointer is moved during step 3, a dotted drag-box is used to indicate the current position of the frame. Moving the pointer around the screen repositions the drag box whilst the shape and size of the frame remain fixed. When the mouse button is finally released, the frame is moved to the new position.

To alter the size and shape of a frame:

1. Point inside the frame

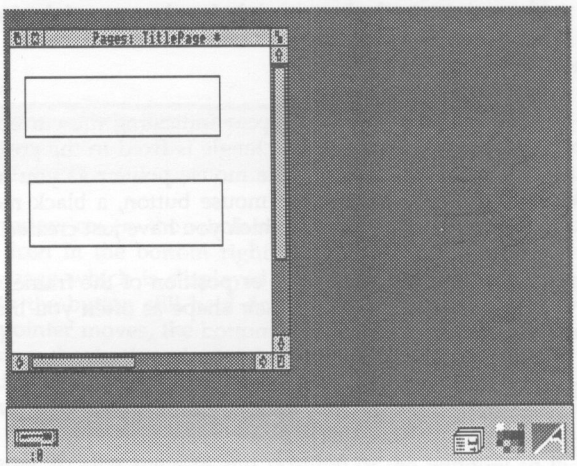
2. Press and hold down the *right* mouse button

3. Move the mouse pointer to reposition the bottom right-hand corner

4. Release the mouse button

Again, a dotted drag box is used to indicate the current size of the frame. In this case, the top left-hand corner of the frame remains fixed. Moving the mouse moves the bottom right-hand corner. When the button is released, the frame is resized accordingly.

Use these techniques to create two frames which look roughly like the ones in the picture below:



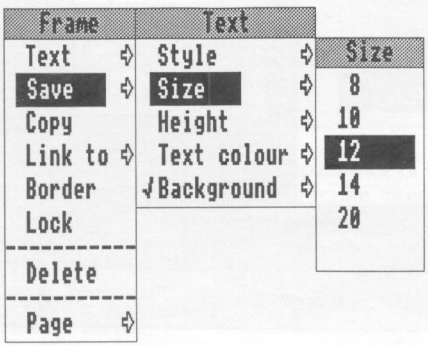
Adding text to a frame

When you have created a frame, you are then free to attach data to it. Various different types of data can be used, which will all be covered in full later. The most important of these is probably text.

To enter text into a frame, click with the left mouse button inside the frame to position the text cursor and then start typing. Words will automatically be formatted so that they are not split between lines. If you make a mistake, you can reposition the cursor by pointing at the position in the text where the error occurs and pressing the left mouse button. Then the delete key can be used to delete the characters, one at a time. Pressing this key on its own deletes the character before the cursor, pressing it whilst holding down the SHIFT key deletes the character to the right of the cursor. Alternatively, extra text can be added at the cursor position by typing in the characters required; the following text will be shifted to the right to accomodate each new letter.

For the purpose of the example being created in this chapter, all you need to type is the word **Archimedes** in the top frame and **Dog** in the bottom. Once you have

done this, there are many ways in which you can alter the appearance of the text. These can be chosen from the menu obtained when you point within the text frame and press the menu (middle) button. The first option is called **Text.** Slide off this entry and a submenu will be displayed showing the characteristics which are under your control; **Style, Size, Height, Text colour** and **Background.** (The last of these has a tick next to it - ignore this for now, it will be explained later.) Sliding off any of these will bring up a further menu showing the possible settings for each. For example:



To select one of these settings, point at it and press the left mouse button. The menus will then disappear and the text within the frame will be redrawn to reflect the change made. Try altering the appearance of your text. For example, alter one frame so that it contains yellow text on a blue background.

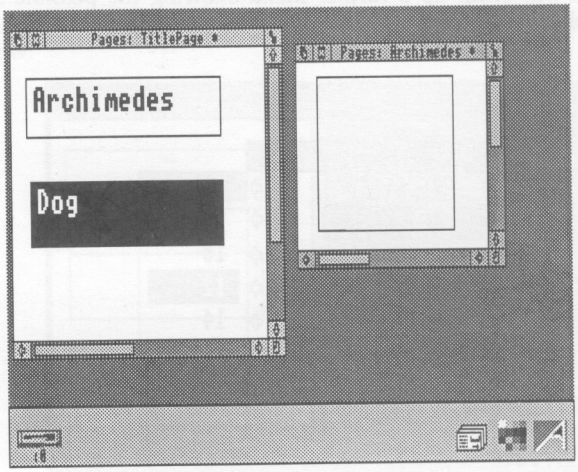
Creating a second page

To create another page, point at the Genesis icon on the icon bar and press the menu button. The menu will now be different to the one originally presented. This is because you now have an application which you are working on and so an application menu is displayed. Before, you hadn't yet created an application so the Genesis editor menu was given. Select the **Create Page** option and another blank page will be displayed.

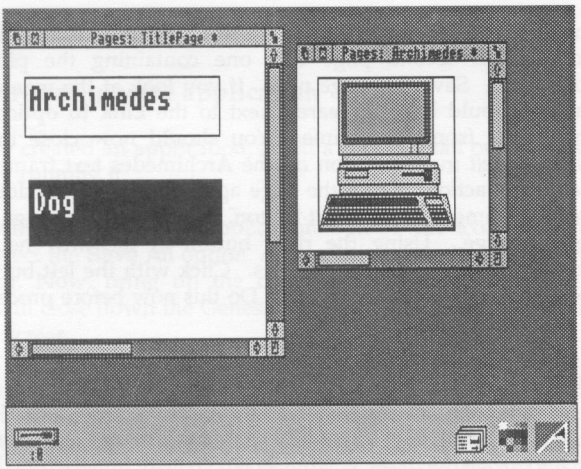
At the top of the page, the title will be shown. This consists of the name of the Genesis application followed by the name of the page, which is currently **<untitled>.** To give it a title, point inside the page, press the menu button and

slide off the **Title** option. A dialogue box will be displayed. Delete the current title using the delete key, type in the new name, for example **Archimedes** and press the enter key.

Now, resize the page, position it and create a frame as described above, so that your screen looks similar to the following:



This time, you should try attaching a different type of data to the frame. The **!GenLib** directory of your Genesis work disc contains a number of draw files, sprites etc for you to use. Open a directory viewer onto your Genesis work disc by clicking on the appropriate disc drive icon. Then double click on the **!GenLib** directory icon. Again you will be presented with a choice of directories. Double click on the **Draw** directory icon and a selection of draw files will appear. Point at the one called **Archimedes** and hold down the left mouse button. With the button still held down, drag the icon until it is over the frame which you created above and release the button. The contents of the draw file will be displayed filling the frame:

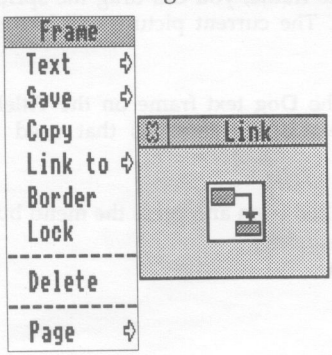


One thing which you can do to improve the appearance of this drawing is to *turn off* the frame border. To do this, point inside the frame, press the menu button and then point at the **Border** entry on the menu. This should be marked at the left by a tick. Clicking on the border entry will remove the frame border. The next time you look at this menu you will see that the border entry is no longer ticked, indicating that the border is turned off. Clicking on this option again will bring back both the border and the tick on the menu.

Linking

Now that you have two pages, you can create a link between them. For example, you might make a link joining the **Archimedes** text frame on the title page to the second page which contains the picture of the Archimedes computer.

First, tidy up your screen by closing any directory viewers you have displayed so that just the two pages are visible. Then, point at the Archimedes text frame and press the menu button. The relevant menu option is called **Link to**. If you slide off this option, a small window containing the **Link tool** will be displayed:



To forge the link, use the left mouse button to drag this tool from the window and drop it on top of the second page (the one containing the picture of the Archimedes computer). Save this page now. If you look at the menu for the text frame again, a tick should have appeared next to the **Link to** option, indicating that a link now exists from that frame. You should now close the page and double-click with the *left* mouse button on the Archimedes text frame on the title page to see the link in action. Close the page again and this time double-click on the Archimedes text frame with the *right* button. This causes the page to open but also *closes* the title page. Using the right button in this way helps to avoid cluttering up the screen with too many pages. Click with the left button over the icon on the icon bar to reopen the title page. Do this now before proceeding to the next section.

Copying pages

Having made a page which is linked to the **Archimedes** text frame on the TitlePage, you may wish to make a similar one which links from the **Dog** text frame. One method of doing this would be to repeat the process described above to create a new blank page, draw and position a frame (or frames) within it and finally attach some data to it. However, a simpler method is to make a copy of the **Archimedes** page.

To do this, point at part of the Archimedes page which is not occupied by the frame and press the menu button to make the Page menu appear. Then select the **Copy** option. After a brief period of disc activity the contents of this page will appear to be redrawn, leaving your screen looking exactly as it did before. However, what has actually happened is that a new page has been created on top of the old one with exactly the same contents.

This page is already a suitable size and has a frame positioned within it. Therefore, all that remains to be done is to change its title to something more appropriate, such as **Dog,** using the **Title** option of the Page menu, and then to change the data attached to the frame. The data you should use this time is the sprite called **Dog** which can be found in the Sprite subdirectory of **!GenLib.**

To change the contents of the frame, you can drag the sprite file icon, **Dog,** from the directory viewer onto it. The current picture will be replaced by a digitized picture of a dog.

Then, create the link from the **Dog** text frame on the TitlePage to the page you have just made. The process is the same as that used above, the steps are summarised below:

1. Point at the frame to be linked from and press the menu button

2. Slide off the **Link to** option

3. Drag the Link icon and drop it on top of the destination page

Saving and running your application

You have now created an application, albeit a rather small one, so you should save it and then try running it.

To save it, point at the Genesis application icon on the icon bar, press the menu button and select the **Save All** option. After a brief period of disc activity the menu will disappear. Now, bring up the menu again, and this time select the **Quit** option. This will close down the Genesis session, remove the icon from the icon bar and clear the screen.

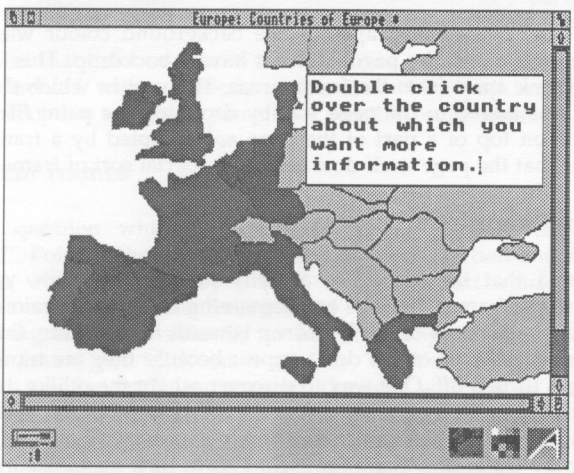
If you open a directory viewer onto your Genesis work disc, the icon, **!Genstart** (or whatever you chose to call it) should be present. Double click on this and the application will start up by displaying the title page. If you then double click on either of the text frames the links you created will be used to display a further page containing the appropriate graphics.

This brief introduction has gone through the basic principles involved in making a Genesis application. Using the same techniques, you could go on to add more pages and data and build up a much more complicated linking structure between them.

4 RUNNING AN EXISTING APPLICATION

This chapter takes you through an existing Genesis application, to demonstrate what can be achieved using the techniques discussed above plus one or two other features of the system. The example being used is **!Europe** which is supplied on the Genesis Example Applications disc. You should ensure that you have made a copy of this disc as described in Chapter 2 *Installation* before proceeding.

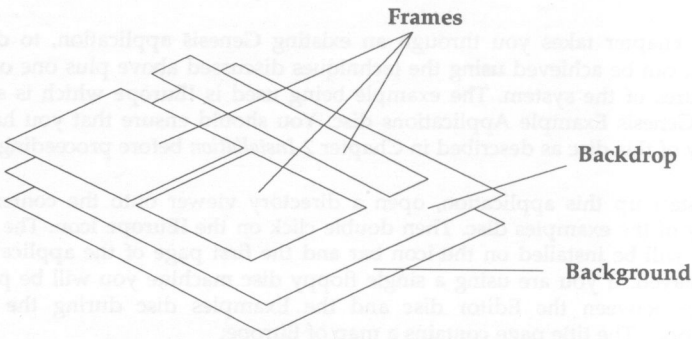
To start up this application, open a directory viewer onto the contents of your copy of the examples disc. Then double click on the **!Europe** icon. The application icon will be installed on the icon bar and the first page of the application will be displayed. If you are using a single floppy disc machine you will be prompted to swap between the Editor disc and the Examples disc during the loading of !Europe. The title page contains a map of Europe:



Backdrops

At first sight, this page may appear to consist of one frame which contains a picture of a map of Europe, with a text frame containing the instructions at the top right. However, this is not the case. The map of Europe is not attached to a frame, it is the **backdrop** of the page.

The contents of a page can be thought of as being in three layers. At the back is the **background,** this can be any single available colour. In front of this there can be a **backdrop.** This can be a drawing, sprite, digitized picture etc which is drawn so that it fills the whole page. Finally, at the front, there can be a number of **frames** with data attached to them.



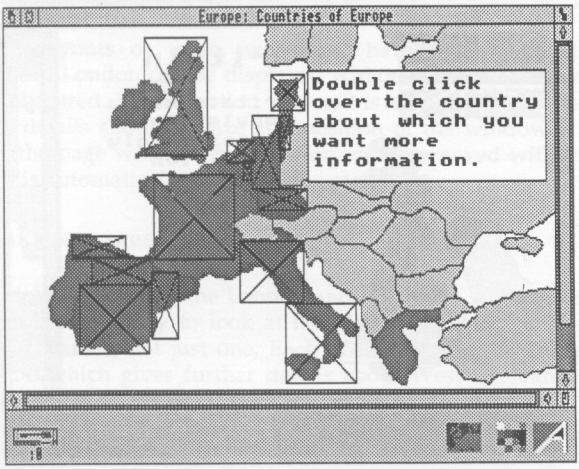
In the example you created in chapter 3, the background colour was set to the default colour of white and the pages did not have a backdrop. This is why they appeared to be blank apart from the frame areas. The way in which the backdrop in this example was added to the page was by dropping the paint file containing the Europe sprite on top of a part of the page *not* occupied by a frame. In some ways, this means that the page itself does act like a special sort of frame.

Locating frames and links

Having discovered that the picture is actually part of the page, you may be wondering where the frames are. The one containing the text is obvious since it is *blotting out* a rectangular area of the backdrop beneath it. However, this is not the only frame which exists - the others don't appear because they are transparent and have their borders turned off. One way to discover where the others are would be to turn on their borders in a similar manner to the way in which you turned off the border of the Archimedes frame in the previous chapter. However, to do this, you need to display the Frame menu for each. This in turn means that you have to know where each frame is to be able to point inside it to obtain its menu. This appears to be a circular problem - you can't display a frame's border because you don't know where it is, but you can't find out where it is without turning on its border! However, a way out does exist.

Point at the bottom right of the map and press the menu button to make the Page menu pop up. If you slide off the **Show** option, you will be given two choices: **Frames** and **Links.** Click on the **Frames** option. Selecting this option makes all the frames on the page visible by displaying their borders in red. You can see that there are several of these, lying over individual countries.

Repeat this action, but this time choose the **Links** option. Now, a red cross will appear in all the frames other than the one containing text:



This indicates that links exist from all the frames except for the text frame. If ever you are presented with a Genesis application which you are not familiar with, selecting this option will show you all the possible routes from the page.

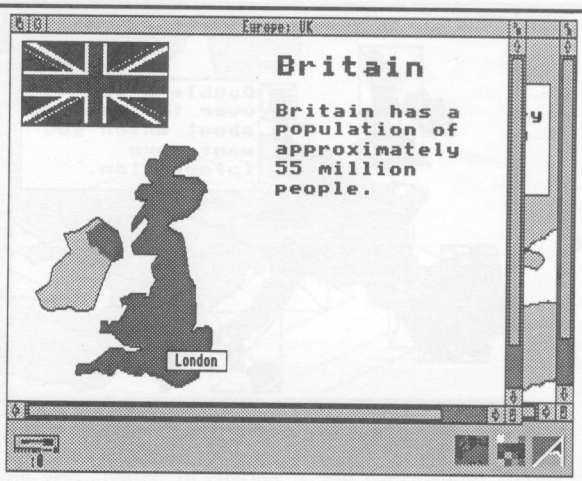
Transparent frames

The other question which you may be asking is '*How can frames be made transparent?*'. Point at the text frame, press the menu button and slide off the **Text** option. This will display the options for editing the text which you have already met. As was noted before, the **Background** option has a tick next to it. This means that the background is visible. If, instead of sliding off the option to reveal the choice of colours, you click on it directly, the tick will be removed and the background will be made transparent. The text will still be displayed but the backdrop will show through it. To make the frame background opaque again, repeat the procedure. Selecting the **Background** option toggles the state ie if the background is opaque, it becomes transparent and vice versa.

Therefore, the frames over the countries are simply empty frames whose borders are turned off and whose backgrounds are transparent. Using this type of frame is a useful technique for adding frames to a page unobtrusively. In this example, the aim is to allow the user to double click over a country to obtain further details. The frames have been made to cover the countries as well as possible.

Selecting a link

Having discovered where the links are, you can now double click inside one of the frames to move onto another page. Try the frame over Britain to start with. This will display a page containing further information about the UK:



This is a good example of a 'typical' Genesis page. It contains a couple of different pictures and some text. Note that the text is in a mixture of sizes. Although within a single frame the text has to be the same font, colour, size etc, separate frames can have totally different characteristics.

This page also has a link to another about London. If you wish, you can see the link by using the **Show ▸ Links** option from the Page menu.

Double click on this text frame with the left mouse button to display the linked page:

Europe: UK
Britain
Britain has a population of approximately 55 million
Europe: London
London
London is the capital city of Britain. It has a population of 8 million people which increases to some 13 million during the working day.
London

The sizes and layouts of these two pages have been planned so that the information about London can be displayed without any of the information about Britain being obscured. This attention to detail is easily achieved because when a page is saved, details about the size and position of the window it is within and the portion of the page which is currently on view are saved with it. Hence when it is recalled, it is automatically shown in the same state.

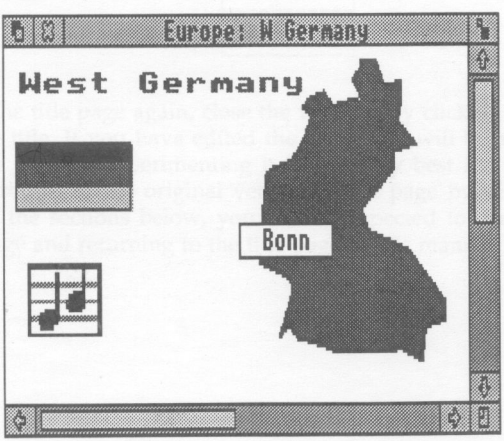
Multiple links to pages

Close the windows containing the London and UK pages so that the original map is visible again. The country to look at next is West Germany. Note that this is covered by four frames, not just one. Each of these frames is linked to the same page, ie the one which gives further details about West Germany. There are no rules about how many frames can be linked to a given page. You can choose to have every link on a page leading to a different destination page. Equally, you can make them all link to the same one. In addition, frames on different pages can be linked to the same destination page. You can even make a link from a frame to its own page, although this is not particularly useful!

In this case, more than one frame has been used to obtain a better coverage of the country. A better fit could be obtained for other countries as well by using several smaller frames to 'tile' the area. From the user's point of view, the application is meant to give the appearance that clicking anywhere on a country will bring up information about that country. However, at the moment, if you were to point at the bottom of France and double click, nothing would happen because there is no frame covering that area. Adding another frame here would improve the illusion which the program is trying to portray.

A different type of data

Double click within one of the frames covering West Germany to bring up the page about that country:



This page contains text and graphics similar to that found on the page about Britain. However, it also contains a new feature.

In the bottom left-hand corner, there is a frame containing the icon for a **Maestro music file**. Like frames containing graphics, this frame can be moved and resized and the contents will alter accordingly so that the icon always fills the frame. Now, try double clicking within this frame - the tune contained in the file will be played.

This type of data is attached to a frame in the same way as a drawing or sprite, that is by dragging the file icon and dropping it into the frame. However, instead of the contents of the file appearing within the frame as would happen with a drawing or sprite, the icon is drawn there instead. The contents of the file are only used when the frame is double clicked on.

Music frames are said to have an *implicit action* associated with them. Double clicking within one will always play the tune contained in the file. This prevents it having any other action. It cannot be used to link to another page. If you point within the frame and press the menu button, you will see that the menu does not contain the **Link to** option.

Further study

This chapter has demonstrated a few more features of Genesis. Try moving around some of the other pages contained in the application. You will find further examples of the different types of data you have met so far. You could also try adding further pages describing more countries and cities or editing some of the existing ones.

5 FURTHER TYPES OF DATA

This chapter uses another example application to illustrate the various types of data which can be attached to frames. The example being used is **!AboutGen** which is also on the Genesis example disc.

To start up this application, open a directory viewer onto the contents of your copy of the examples disc. Then double click on the **!AboutGen** icon. The application icon will be installed on the icon bar and the first page of the application will be displayed. The first page contains the names of several different types of data. Each of these is in a separate frame which can be double clicked on. This fact can be illustrated by selecting the **Show ▸ Links** option from the Page menu. This will highlight all the frames which have links from them.

To obtain further details of any of the types, point at the one you are interested in and double click with the left mouse button. This will display a further window giving a description of the data type and one or more examples of its use. The Frame menu for any of the example frames can be obtained by pointing within the frame and pressing the menu (middle) button. In each case, the first entry on the menu will be specific to the type of frame. For example, if the frame contains text, the first menu entry will be **Text:**

To return to the title page again, close the window by clicking on the Close icon to the left of the title. If you have edited the page, you will be asked if you want to save it. While you are experimenting it is probably best for you to discard your changes and return to the original version of the page by selecting the **NO** box. After each of the sections below, you will be expected to carry out the action of closing the page and returning to the title page in this manner.

Text

The first, and most commonly used, data type is text. You have already encountered several examples of text frames. However, it is well worth examining them again since they are so important.

If you double click on the 'Text' frame, you will be presented with a small page showing text frames in use. Text can be added to frames in two ways. Either it can be typed directly into the frame, or a text file can be dragged and dropped on top of a frame. The result, whichever way the frame is created, is the same.

Text frames are different from other types of frame in that they can be edited directly. To see how easy this is, click with the left mouse button after the final word on the screen. A red caret will appear marking the current text position. Then try typing the sentence '**In addition, the background colour can be altered**'. The first part of this sentence will appear on the screen, filling the remaining space on the final line. However, when you reach the end of the line, no more characters will be accepted because the frame is full. Text is not scaled so that it fits the frame.

To enter the whole amount, you can increase the size of the frame. To make this easier, you should first turn the frame border on. Point at the main body of text, press the menu button and select the **Border** option. A black border will be drawn around the edge of the frame. Then point within the frame and press and hold down the right mouse button. This allows the size to be adjusted. Gradually, move the mouse pointer down towards the bottom of the screen and the frame will expand. When you have moved it by about a centimetre, release the mouse button. The frame will be redrawn and you can finish typing the sentence.

Font size

An alternative method of fitting more text within a frame is to reduce the size of the lettering. To demonstrate this, you should first close down the Text window. Click on the close icon and when asked if you want to save your edited page, reply by clicking on the **NO** box. If you then open the page up again by double clicking on the 'Text' frame, it will appear in its original form, without the edits you made. Then display the Frame menu and slide off the **Text** option to obtain the submenu.

To decrease the overall size of the characters, slide off the **Size** option. This gives a submenu from which you can choose a number of different text sizes. The current value, in this case 14, is marked by a tick. Select a smaller size by clicking with the left mouse button. Alternatively, you may have noticed that there is a **Height** option. This operates in the same way but only changes the height.

You should now have plenty of room for the extra sentence.

Colours

Similarly, you can change the colour of the text or the background colour within the text frame by sliding off the **Text colour** or **Background** options and selecting the shade required from the menu provided. In many cases, making the background colour of the text frames the same as the background colour of the page itself produces the most pleasing effect. If the colours are different, the frames stand out from the page and appear to be isolated blocks.

If you click on the **Background** option itself (rather than sliding of it) then the background becomes transparent. This can be very useful when writing captions for diagrams etc.

Drawings

On the Drawings page are two examples of frames containing draw files. These can be produced by Draw, the drawing program supplied with all Archimedes machines and many other externally available applications such as Presenter 2 from Lingenuity.

To attach a draw file to a frame, you just drag the appropriate draw file icon and drop it on top of the frame. The contents of the file are then scaled to fit within the frame. Point at one of the examples given, press the menu button and slide off the Draw option. This gives you the name of the draw file from which the drawing came and the size of this file. There are no editing options available. If you want to alter a draw file you have to return to Draw or a similar application, make the changes you want, and replace the old version of the file attached to the frame with the new edited one.

Drawings can come in many shapes and guises. They can consist of pure line drawings or solid shaded pictures as shown here. Note that drawings can contain text. The graphical example created using Presenter 2 contains text for marking the axes, giving the colour key etc. However, because it is part of the drawing it cannot be edited within Genesis.

Another feature of drawings is that any area of them which is not explicitly coloured in, is transparent. To illustrate this more clearly, display the page menu by pointing at part of the page not occupied by frames (for example the top right corner) and press the menu button. Then slide off the **Background** option and select colour 8 which is royal blue. The text frames retain their grey background because this has been set explicitly (their backgrounds have not been made transparent). However, the computer drawing is now surrounded by blue instead of grey. This is because the background of the draw frame is transparent so you can see through it to whatever is beneath.

Sprites

Sprites are a different form of graphical object. They are produced by painting programs such as Paint which is supplied with the machine and other externally available ones such as Artisan from Clares. In addition, digitizers which create digitized pictures of objects store these images as sprites.

The Sprites page contains two frames containing sprites, a simple one created using Paint and a highly detailed digitized image. Like drawings, sprites are attached to frames by dropping the appropriate file icons on top of the frames. Note that a sprite file created using Paint can contain a number of different sprites. Genesis will use the first of these and will ignore the rest. This is similar to the behaviour of other applications. Therefore, it is sensible when using Paint, to store sprites individually in separate files. This has the added advantage that the file name can be used to more clearly indicate its contents.

Sprites are also similar to drawings in that they cannot be edited directly in Genesis. Sliding off the **Sprite** option contained in the Frame menu of these two examples again just gives information about the name of the sprite and its size.

Maestro music

The Maestro music editor which is supplied with Archimedes computers can be used to create music files. These in turn can be dragged and dropped on top of frames to attach the music file to the frame. In this case there is nothing graphical within the file which can be displayed within the frame. Therefore the icon representing the file is drawn there to make it obvious what type of frame it is.

The Music page contains one example of this type of frame. To hear the music played, double click on it. If you display the Frame menu and slide off the **Music** option you will be presented with information about the name of the piece of music and its size. In addition, a **Stop** option is given. If you click on this, the tune will stop immediately.

Maestro files can be used to good effect to make applications more interesting. For example, the !Europe application contains a couple of Maestro files. Instead of just describing each country using text and graphics, music is used in some cases to demonstrate national anthems. This provides variety and increases the user's involvement since he/she has to activate the frame to hear the music rather than just passively looking at the page contents.

This feature also allows specialist musical databases to be compiled. For example a database of composers is not limited to just giving details of their life and works, it can also include musical fragments to demonstrate the type of music they wrote.

Similarly, a catalogue of different instruments can contain examples of the type of sound they produce.

If your Archimedes is linked to any musical equipment via the standard MIDI interface, then any sounds which Genesis produces from Maestro files will be output through this equipment.

Euclid 3D pictures

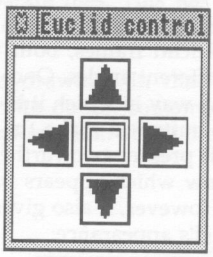
Euclid is a separate package available from Ace Computing which allows three dimensional pictures to be created. These pictures can be attached to frames in the usual manner. Although you need the Euclid package to create pictures of your own, Genesis allows you to display existing pictures without you having to have the Euclid editor.

The Euclid 3D page contains two Euclid frames, both with the same 3D picture attached to them but viewed from different angles. Once you have a Euclid picture attached to a frame, you can alter the way in which it is displayed. Point at the left hand Euclid frame, press the menu button to make the Frame menu pop up and slide off the **Euclid** option. This will present two further choices: **Info** and **Panel.** Slide off the **Info** option. The window which appears provides information about the name of the object and its size. However, it also gives two sets of radio buttons which can be used to select the model's appearance:

The first set of three allow you to choose between **Wire frame** drawings in which just the edges of the model are displayed, **Faces only** in which the separate solid blocks of the model are shown but the joins between them are not and **Faces & edges** in which both are visible. The last of these is the mode currently being used and is the most effective of the three. However, try selecting the wire frame mode by clicking in the button to its left and then on the **OK** box to see the effect produced.

The second set select the lighting to be applied to the model. The type of lighting only has an affect when faces are visible. Return to the same menu again and click in the **Faces & edges** button. Then also select the **Front light** option before clicking on **OK** to apply these changes. This will cause the solid model to be displayed as though it is being illuminated from the front. Those parts of the model facing towards the light source will be lighter than those facing away. The other option, **Lights on**, illuminates the model using lights which were placed when the model was created.

Bring up the Frame menu for this frame once more, slide off the **Euclid** option and this time select the **Panel** option. This will make the Euclid control panel appear. Using this panel, you can alter the size, position and orientation of the model.



The centre box controls the size. Try clicking once in the centre box with the left mouse button, this will make the model become larger. Now do the same with the right mouse button and it will become smaller again. If you hold down one of the buttons whilst pointing at this box rather than clicking and letting go, the picture will zoom in or out. Then as soon as you let go, it will stop at the current size.

The arrows affect the positioning within the frame and the orientation. Try clicking on these arrows with the left mouse button. You should find that the model moves relative to the frame either up, down, left or right depending on the direction of the arrow chosen. Now repeat the action but this time, hold down the SHIFT key at the same time. Instead of the model moving sideways, it will rotate. The left and right arrows cause it to rotate sideways, the up and down arrows make it rotate so that you appear to be moving over or under it. Using these it is possible to create plan views and different elevations of a model.

To practice using these actions, try to make the view of the model in the left-hand frame, the same as the view in the right.

Some Euclid models are provided in the **!GenLlib** library for you to use in your own applications. However, it is well worth investing in a copy of the software so

that you can create ones of your own. 3D is a very powerful visualisation tool which is becoming increasing popular.

Mogul Films

Mogul is a separate package from Ace Computing which allows 3D objects to be animated. The Mogul Films page gives an example of a Mogul frame which has attached to it a file containing an animated sequence of the five Olympic rings.

Point within the frame, press the menu button and slide off the **Film** option. This provides the usual information and in addition gives three options: **Pause, Play** and **Step.** Try clicking on **Pause,** the animation will stop and the contents of the frame will be frozen in their current state. Now, repeatedly click on **Step.** This will take you through the individual stages of the sequence one at a time. When you want the animation to continue again, click on **Play.**

Mogul frames are often the highlight of a Genesis application, particularly if it is intended for young children. However, be warned that each film can potentially use up a large amount of memory and disc space so they should be used sparingly.

Other datatypes

You have now seen all the types of data which Genesis recognises and treats in a special way. However, there are many other types which have not been mentioned, so what happens if you try to attach one of these to a frame? The answer is that Genesis provides a default action for all types of data which it does not recognise.

When an unknown datatype is dragged to a frame and dropped on it, the visible effect is that the icon is drawn within the frame. In addition, an appropriate action occurs when the frame is double clicked on. The action is the same as that obtained when the icon is double clicked on within a directory viewer.

For example, create a frame in the bottom left corner of the title frame which is roughly two centimetres square. Then open a directory viewer onto your Genesis work disc and double click on **!GenLib**. When the contents of the resources directory have been displayed, locate the Draw directory icon and *drag* this icon onto the frame you have just created. This places the directory icon in the frame. Normally, if you double click on a directory icon within a directory viewer, another directory viewer will appear displaying the contents of the second directory. Therefore, this is the action which should occur when you double click on the frame. Try this now, for confirmation.

Similarly, if you use an icon representing an application such as PipeDream, then double clicking on the frame will start up the application. For the icon of a

PipeDream document, PipeDream will be started up and the document will be loaded into it ready to be edited. If you have bought other applications for your Archimedes which create files of a particular type then try this out now for yourself.

Words

There is one final type of frame which can be created in Genesis called a *word frame*. This is a very simple type of frame which doesn't have a file attached to it. Instead, it contains a short string of characters which is stored directly as part of the frame definition. It can contain a single row of characters which are displayed in the centre of the frame. If the frame is not wide enough to hold all the characters then the string will be clipped to the sides so that as many characters as possible are displayed. These characters can only be drawn in the standard system font, their size cannot be altered.

Every frame which you create starts out as a word frame, this is the default type. To see this, select an empty area of the page and create a frame in it. Then display its Frame menu before clicking inside it. You should see that the first entry is **Word.** Sliding off this displays the options which are available for this type of frame: **Word, Word colour** and **Background.**

Word is the option provided to allow the characters to be entered or changed. Sliding off this option will display a dialogue box containing the current contents of the frame. You can move around this using the left and right arrow keys. Pressing the delete key on its own will delete the character before the cursor, pressing the COPY key deletes the characters to the right of the cursor. If you type a character, it will be added at the current cursor position, all characters to the right will be moved along to make room for it. To update the frame so that it reflects any changes, press the enter key. At any time before this, you can click with any of the buttons elsewhere on the screen to remove the menu and the changes will be forgotten. Try out this option, to place the string 'Hello' in the frame you have just created.

Word colour and Background give a choice of colours for the text or background. The background can also be made transparent by selecting the Background option directly.

A newly created frame tends not to remain as a word frame for very long. If you click in it with the left mouse button then it automatically changes into a text frame so that characters can be entered. This happens even if you don't actually enter any text. Even clicking within the frame in order to move it around the page is sufficient to convert it. If you wish to make the frame remain as a word frame, it is important to enter a word string into it as soon as it is created. Once it contains such a string it is safe from accidental conversion.

The value of word frames will be shown in the next chapter.

6 DATABASES

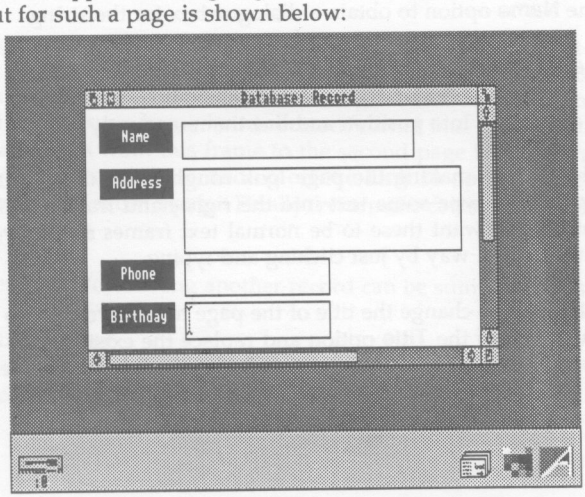
One of the many uses of Genesis is for creating databases. This chapter considers the simple example of an address book which stores names, addresses, telephone numbers and birthdays.

Creating the first page

The first thing you need to do is create a new, empty Genesis application which will become your address book. The way to do this was explained in Chapter 3, *Creating an Application*. However, to refresh your memory, the stages are listed briefly below:

1. Obtain a directory viewer onto your Genesis work disc by clicking on the disc icon on the icon bar
2. Double click on the *!Genesis* icon - this starts up the Genesis Editor and installs its icon on the icon bar
3. Point at this icon and press the menu button to obtain the Genesis Editor menu
4. Slide off the **New** option to display a window for creating a new application
5. Delete the contents of the dialogue box using the delete key and replace them by the name you want to give to this application, for example **!Database**
6. Finally, drag the application icon and drop it on the directory viewer to create the application and obtain the first blank page

Each page of this application is going to hold the data for a different person. The typical layout for such a page is shown below:



The frames on the left hold the titles of the pieces of information about a person which are going to be entered into the frames on the right.

Before you go ahead and create these frames, there are a couple of points to note. For neatness, the left-hand frames have all been created the same size. An easy way to achieve this is to create the first frame in the normal way using the mouse button and then take copies of it. To save time, you should change the colours of the text and background for the first frame before you copy it. This means that the copies will be created so that their colours are set up correctly. To do this:

1. Point at the empty frame and press the menu button

2. Slide off the **Word** option

3. Slide off the **Word Colour** option and select Colour 0 (white) for the text

4. Repeat steps 1 and 2 then slide off the **Background** colour option and select colour 7 (black) for the background

You can then make each copy by obtaining the Frame menu for the first frame and selecting the **Copy** option from it.

The other point is that the left-hand frames must be word frames. It is therefore important that you place some text in them before you try to move them, otherwise they will get changed into text frames when you click on them. To set them up:

1. Point at the empty frame and press the menu button

2. Slide off the **Word** option

3. Slide off the **Name** option to obtain a dialogue box for the string

4. Type in the contents of each frame and press the enter key

You can then drag them into position and line them up neatly

Now create the frames, making the page look roughly like the picture shown on the previous page and type some text into the right-hand frames which describes you. Note that you do want these to be normal text frames so that you can enter text in them in the usual way by just clicking and typing.

The finishing touch is to change the title of the page to 'Record'. To do this, display the Page menu, slide off the **Title** option and replace the existing title displayed in the dialogue box by this name. When you are happy with it, press enter.

Possible problems

There are two possible problems which you may encounter whilst creating the frames. The first is that you accidentally change one of the left-hand frames into a text frame by clicking in it with the left mouse button while it is still empty. If this happens then there is no way in which it can be changed back again. You have to delete the frame using the **Delete** option of the Frame menu and then recreate it again either from scratch or by copying an existing word frame.

The second is that the text does not appear in the right-hand text frames when you try to type it in. This will be because the frame is not high enough to display a whole character. You then have two choices, you can either increase the height of the frame using the mouse or decrease the height of the characters by selecting a new size from the **Text ▸ Height** option of the Frame menu.

Subsequent pages

One of the main features of a database is that the pages contain the same pieces of information using the same layout. This makes it easier to locate any item of data that you are interested in. With traditional databases you are often restricted to having to use the same layout. Genesis is more flexible in this respect in that it allows you to alter any of the pages if you want to hold extra information about somebody. However, you should try to keep them the same if possible.

The easiest method of keeping all the page layouts the same is to copy them. You've already encountered one way of doing this which is to use the **Copy** option from the Page menu. Try this again now. Point at part of the page not occupied by a frame, press the menu button and select the **Copy** option. An exact copy of the page will be created and displayed on the screen. You can now edit each of the text frames in turn to delete the current contents and type in the details for a second person.

The other action you need to perform is to make a link from the first page to the second so that you have a means of moving through the pages. Reduce the size of the current page window so that the bottom right-hand corner of the first page below it is visible. Create a frame in this bottom corner and type the phrase **go on** in it. Then create a link from this frame to the second page by pointing within the frame, pressing the menu button, sliding off the **Link to** option and dragging the Link tool on top of the second page. Finally, return the second page window to its original size.

The process described for making another record can be summed up as follows:

1. Copy the current page

2. Delete the contents of the text frames

3. Type in the details for the next person

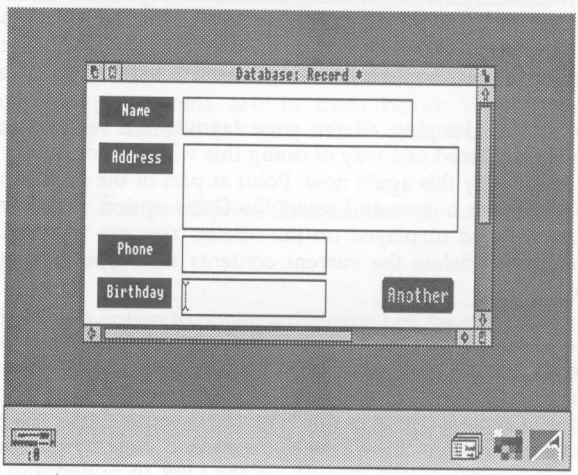
4. Form a link between the previous page and this one

Although this is a fairly easy process, Genesis provides the means of automating it even further. This is examined in the next section.

The 'Another' button

Up to now, whenever you have wanted to add something to a page, you first of all created a frame and then attached data to the frame. However, it is possible to place an existing frame plus its attached data and action information on the screen all in one go. The word **button** is used to describe this combination of frame, data and action information.

Some predefined buttons exist in the buttons subdirectory of **!GenLib.** Open a directory viewer onto this directory and drag the **Another** icon to your page, placing it in the bottom right-hand corner. As soon as you drop the icon, the button will be displayed on the page as follows:



This button has a special type of action associated with it. Previously, the only actions you have encounted have been links to other pages or the implicit actions belonging to frames with data such as Maestro music files attached to them. However, by using some of the more advanced features of Genesis, more sophisticated actions are possible.

To see part of the action of the Another button, try double clicking on it. This will cause a new page to be created and displayed. The contents of this page will be exactly the same as the previous one, except that the text frames will be blank (note that the Another button itself is also copied). Therefore this button has already cut out one of the steps outlined in the copying process since you no longer have to delete the contents of the text frames yourself!

It is only the text frames which are cleared by the action of this button. The word frames are copied exactly as they appeared on the original page. This is why it was emphasised above that the database's field names had to be words. If you had made them into text frames they would have been cleared as well.

The second part of the action of the Another button is to make a link from itself to the new page. Finally, it converts itself into a **Next** button.

To see this for yourself you should first save the application by selecting the **Save all** option from the application icon and then close down the application by selecting the **Quit** option.

Start up the application again by double clicking on the **!Database** icon. The first page will automatically appear, exactly as you created it. Double click on the **'go on'** frame in the bottom corner to bring up the next page. This will display the second page and you should see that the original Another button has changed so that it now contains the **Next** sprite.

To prove that the button has changed in action as well as appearance, double click on it. The third page will be displayed. Note that the Another button in this page is still there in its original form ready to be used to create further new pages.

So the actions of the Another button are:

1. Create a new page exactly like the current one but with the contents of any text frames blanked

2. Make a link between itself and the new page

3. Change into a **Next** button

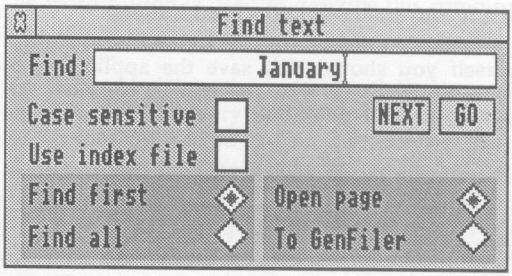
This gives you precisely what you want. The last page in any application made in this way will always contain the Another button to allow another blank record card to be created. However, previous ones will contain the Next button which will take you onto the next existing record. You don't have to worry about copying pages or making links, all this is taken care of for you automatically.

More information about buttons appears in later chapers.

Searching

With any Genesis application, but especially the database kind, it is useful to be able to locate certain items of text quickly. For example, with the database you have just created, you might want to find out who has a birthday coming up in the next month. One slow way of doing this would be to look through every page of the application. However, to speed things up, Genesis contains a *find* facility.

Display the application menu and click on the **Find** option. The **Find text** window which appears contains a writable icon for the text you want to find and various other option boxes which you should ignore for now. Choose a month and enter it in the writable icon:



Then click on the **GO** box. What happens now is that each page of your address book is searched to see if the text you have entered appears on any of the pages. As soon as a page is found, it is displayed on the screen and the search stops. If the text isn't found, nothing happens. Having found one page, you can continue the search for further pages by displaying the Find text window again and this time clicking on the **NEXT** box. Note that the text you are searching for is still in the dialogue box - it is remembered so that you don't have to re-enter it each time.

This application isn't really large enough to demonstrate the find facility properly. Therefore at this stage you should quit this application and start up **!Europe** from the Examples disc again. When the first page has appeared, display the **Find text** window, type **LONDON** into the dialogue box and click on **GO.** The first page displayed will be the UK page which has a text frame marking the position of London on the map of the country. Now display the **Find text** window again and click on **NEXT.** This will find and display the page about London. If you now try clicking on **NEXT** again, nothing will happen because no other pages mention London.

Close down the London and UK pages and try searching for LONDON again. However, this time, click on the **Open all** pages box before you GO. This causes all the pages containing the text to be displayed in one go - the search does not stop

as soon as the first one is found. You should therefore find that both the UK and London pages have appeared.

Note: there is a limit to the number of pages which can be open at once. By default this is 16. Once the limit is exceeded, 'older' pages will be closed down to make room for the new ones.

Close down the two pages again and try one final search for LONDON having clicked on the **Case sensitive** box beforehand. This time nothing will be found. This is because the option makes the case of the text important. Only pages containing the word LONDON in capital letters will be found. Because all occurences of it are in the form **London** with just an initial capital they are ignored. When this box is empty, any case is accepted. Therefore normally LONDON will find **LONDON, london, London, LONDon** etc.

For most applications, you should leave the **Case sensitive** option off. Usually you will only be interested whether the text appears or not, the case of it will be irrelevant. However, there may be occasions when this is needed. For example, when creating an application you may decide to highlight particular words by entering them in capitals when they are an important part of the text. You can then look just for these occurrences and ignore the lower case, less important ones.

In order to understand the other options, **To GenFiler** and **Use index file** please refer to chapter 9, *The Genesis Filer* and Appendix D, *Index Files.*

Summary

As you have just seen, Genesis does not have a special mode of operation for creating databases. This type of application is treated in exactly the same way as any other. It is only *different* because you may choose to impose your own rules to it, such as making all pages use the same layout. Therefore, methods of duplicating pages are more important to databases than to applications such as !Europe in which each country description was created differently to make it more interesting.

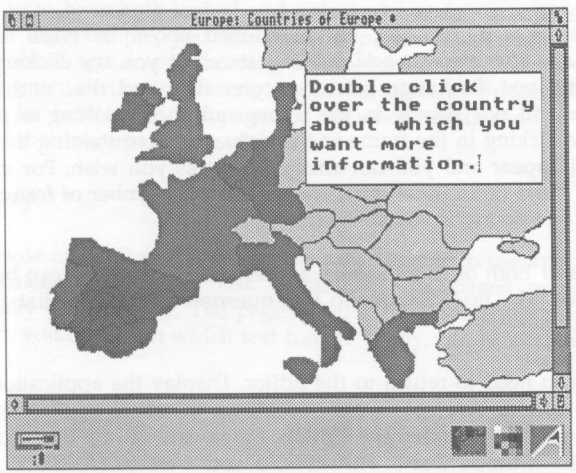
In addition, databases are typically used for the storage and fast retrieval of data. Therefore search facilities are particularly important to them. Again, contrast this to !Europe, in which the idea is for the user to browse through the pages rather than find just one of them as quickly as possible.

7 BROWSING

So far, you have always used the Genesis Editor when working with applications. This has been the case whether you have wanted to create a new application, alter an existing one or merely examine one without changing it. However, the editor is really provided for the first two of these activities; creating and editing. If you just want to examine an application, the Genesis Browser is a better tool to use.

Starting up the Browser

The Browser is supplied on the Genesis Editor disc. To start it up, open a directory viewer onto your copy of this disc and double click on the **!Browser** icon. This will install the icon onto the icon bar. Then, change to the Examples disc, open a directory viewer onto this and locate the **!Europe** example. This example can be started up using the Browser instead of the Editor by dragging the !Europe icon from the directory viewer and dropping it on top of the Browser icon on the icon bar:



At first sight, nothing appears to be different. The initial map of Europe is displayed in the same way and if you try double clicking on a country such as France, the page containing further information about the country will appear, proving that the links are still present.

Editing v Browsing

The difference between the Editor and the Browser is that the latter does not allow you to perform certain editing operations. To demonstate this, double click on France to display its page. Then try changing the size of the frame containing the

French flag. Nothing will happen! The Browser does not permit you to alter the layout of a page in any way - existing frames cannot be moved or resized and new ones cannot be created. In addtion, you cannot change the resource which is attached to a frame. If you try dropping a different sprite file on top of the flag, it will just be ignored.

Now try pointing at the flag and press the menu button. You will see that the Frame menu is much reduced in size. Options such as **Delete, Link to** and **Border** which were present in the Editor are missing.

Similarly, examine the Page menu. This contains only options which do not alter the page contents in any way. The **Show** option is still available to help you locate frames and links because this only highlights the frames in a temporary manner. The red rectangles and crosses which it produces are not part of the real graphics on the page and are not saved.

Frame locking

One question which was deliberately not mentioned above is '*When using the Browser, are you able to alter the text held in text frames?*' If you try clicking in the frame containing the text **Exchange rate** then you will find that nothing will happen. The cursor will not appear in the frame and so no editing is possible. However, if you try clicking in the frame to the right of this containing the text **10** then the cursor will appear and you can alter this text as you wish. For example, try changing it to a more accurate number for the current number of francs which can be obtained for one pound if you know this value.

So what is happening? Both of these frames are text frames but one can be edited and the other cannot. The brief answer to this question is that the first frame is *locked* and the second one isn't.

For further details you need to return to the Editor. Display the application menu and select the **Quit** option. You now want to start up the same application again but this time making it run under the Editor. Previously to do this, you have double clicked on the **!Europe** demo. However, if you were to try this now, you would find that the Browser would be started up again. The tool chosen for running an existing application when it is started up in this way is always the most recently used one. However, if neither the Editor nor the Browser have been used in this session then the Editor is chosen by default.

In this situation, you have to start up the application by double clicking on **!Genesis** to install the Editor icon on the icon bar and then dragging the **!Europe** icon and dropping it on top of the Editor icon. This specifies that the Editor is to be used.

Now double click on France to return to the same page that you were looking at

before. Point in turn at each of the two text frames which you were investigating and display their Frame menus. You should notice that the **Lock** option for the left-hand frame has a tick next to it and the right-hand one has not.

By default, all frames are unlocked. However, you can lock any frame by selecting the **Lock** option on the Frame menu. A tick will then be placed next to the option on the menu. When you are running the Editor, locks can be useful for preventing accidental changes. Locking a frame prevents you from altering it in any way. You cannot change the data within the frame, resize, reposition or delete the frame or even change its link. However, the lock can always easily be overcome simply by clicking on the **Lock** option again to de-select it.

Under the Browser, only the locking of text frames to prevent the text being altered is relevant - all other edits are prevented automatically. However, this one feature is very useful. Since the Browser does not provide the option of *unlocking* a frame, the lock state predetermines whether text frames can be edited or not.

The ability to have both locked and unlocked text frames makes the Browser very powerful. For example, you can create applications for children containing draw frames, locked text frames holding instructions and unlocked text frames so that they can enter their own description of the pictures presented. Save options are available for both individual pages or the whole application so that they can keep a copy of their own personalised version.

Summary

The purpose of the Browser is to allow other people to use applications which you have created without the risk of any of the pages being changed, either accidentally or otherwise. The page layouts, linking information etc remain fixed. However, you can select which text frames, if any, can be altered.

8 THE COMPLETE PICTURE

You have now met most of the components of a Genesis application and have used many of the menu options available. This chapter goes through them all again in order, adding more details where necessary. Reading once through the descriptions below will provide an overview of the whole system. You can then refer back to individual parts of this chapter when necessary to gradually learn the details.

Pages

Pages are the fundamental building blocks of a Genesis application.

A new blank page can be created using the application icon menu. Each page is the same size as the screen. However, it is always displayed in a window whose size can be altered. If the window is shrunk, then only part of the page is visible. In this case, the scroll bars at the side and bottom of the window can be used to scroll around the page and view the data inside different parts of it.

When a page is saved, the information about the portion of it which is currently on view and its position on the screen is saved with it. This has the advantage that displaying one page won't necessarily completely obscure the previous one. The page windows can be used to ensure that one page appears well placed with respect to the previous one. For example, the following shows two pages:

Europe: UK
Britain
Britain has a population of approximately 55 million
London
Europe: London
London
London is the capital city of Britain. It has a population of 8 million people which increases to some 13 million during the working day.

The majority of the one behind is being displayed. This has a blank space in the bottom right hand corner where the second page window will appear if requested. Only part of this second page is shown. It may be that the rest of the page is blank. Alternatively it could contain further information which would be revealed by scrolling or increasing the size of the window.

Once a page has been saved, it will re-appear in the same state when opened. Hence it is not only important to design the layout of individual pages but also to think about how pages are to appear relative to one another.

The contents of a page can be thought of as being in three layers. At the back is the *background*, this can be any single available colour. In front of this there can be a *backdrop*. This can be a drawing, sprite, digitized picture etc, in fact anything stored in the same format as a draw or paint file. Finally, at the front, there can be a number of *frames* with data attached to them.

Frames

Frames are the rectangular areas contained within pages which can have data attached to them.

A frame is created using just the mouse button to drag open a rectangle to the size required. Once created, a frame can be moved around the screen and resized. One frame is allowed to overlap with another, in which case the one which was created later will appear on the top.

Different types of data such as text, graphics, Maestro music files etc can be attached to a frame. In the majority of cases, the data is in the form of an existing resource such as a draw file. This can be attached to a frame by dragging the icon representing the resource and dropping it on top of the frame. The action then depends on the file type:

For draw files, sprite files, Euclid files etc, the graphics contained in the file will be scaled and positioned so that it fills the frame.

For text files created using Edit, or other word processors, the text will be placed within the frame, correctly formatted. It will be printed in the standard system font and will not necessarily all fit within the frame. If this is the case, just the top part of the document will be displayed. More of the text can only be viewed by either increasing the size of the frame or decreasing the size of the characters.

For Maestro files and file types which Genesis does not handle explicitly, the file icon will be drawn within the frame. Double clicking on the frame will cause an appropriate action to be carried out. In the case of Maestro files, the music will be played. In all other cases, the effect is the same as double clicking on the original icon within a directory viewer. For example, if the icon is that of a BASIC program then the program will be run, in the case of a directory icon, a directory viewer will be displayed. If the icon represents an application such as PipeDream then the application will be started up. If the icon represents a PipeDream document then PipeDream will be started up and the document will be loaded into it.

There are two exceptions to the above method of attaching data to frames. The first is the method of producing text frames by typing text directly into the frame. This creates a new text file attached to the frame in which the text is stored.

The second is to create a *word frame*. This contains just a short string stored directly as part of the frame rather than in a separate data file. This string is displayed on one line in the system font centred within the frame.

Links

Links are used to form connections between pages.

To create a link, you need to have the frame which is to be linked from and at least part of the page which is to be linked to visible on the screen. Then, popping up the Frame menu and sliding off the **Link to** option will display a Link tool. Dragging this tool and dropping it on the target page creates the link.

Whenever a frame has a link from it, double clicking within the frame will display the page it is linked to. If the left mouse button is used for this, the previous page will remain on the screen. If the right mouse button is used, the previous page will be closed first and then the new one will appear. This can help reduce screen clutter.

It is also possible to make a link by dragging the page from the Genesis Filer window and dropping it over the frame. This method can also be used to link to pages in *other* Genesis applications.

Buttons

Button is the term used to describe a frame and its associated data and link information. The definition of a button can be extracted and saved separately in a special type of file. These can then be included in other pages. For more information see chapter 10, *Script Files*.

Instead of describing a page as containing a number of frames which have data attached to them and links leading from them to other pages, a page can be said to contain a number of buttons. Double clicking on a linked button, displays the page it is linked to.

Backgrounds and backdrops

Backgrounds and *backdrops* occupy the area of a page behind the frames.

The area of a page which is not occupied by frames does not necessarily have to remain blank. Each page has a background colour which is used to colour in this area. By default, this is white but it can be set to any other colour available using

the Page menu. In addition, a page can have a *backdrop*. This is a drawing or picture which is drawn on top of background and which fills the whole page. Since the backdrop is drawn to fit the whole page, if only part of the page is being displayed due to the page window not being the full screen extent, only part of the backdrop will be visible. In addition, any frames contained in the page will *blot out* the rectangular areas of the backdrop which they lie on top of.

A backdrop can be added to a page by dropping a draw or paint file on top of any area of the page not occupied by a frame. In some ways, this means that the page itself acts like a special sort of frame.

Script files

Script files hold the textual descriptions of pages.

When a Genesis application is first created, an application directory is made for it. For each page of a Genesis application, a new subdirectory is used. Contained within the subdirectory are all the files associated with the page, for example copies of any draw files used, text files for each text frame created and a *script file*. This is a text file which can be examined using Edit if you so wish. It contains, in text form, the description of the page including the sizes and positions of all the frames, link information etc.

More adventurous users may like to experiment with editing these script files since this provides a quick method of altering a page to get exactly what you want. However, there is no need for any user to look at these files if they don't wish to. They are only mentioned here because a knowledge of their existence helps to explain why certain parts of the system work in they way that they do. For more information see chapter 11, *Genesis Script Language*.

Menus

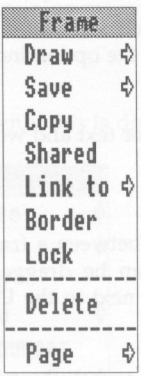
Editing and other actions can be performed on frames, pages and applications by selecting options from the menus obtained from them.

Operations can be performed on complete applications, single pages or individual frames. Each has a menu associated with it which can be displayed by pressing the menu button whilst pointing at a particular place; within the frame for the *Frame menu*, within a page outside of a frame for the *Page menu* and at the application icon on the icon bar for the *Application menu*. The Genesis Filer also provides some menus which are described in chapter 9, *The Genesis Filer*.

Frame menus

The entries within a frame menu depend on the type of frame. For example, a text frame will produce a menu which allows the text to be resized, drawn in a different font etc. This doesn't appear in other frame menus since it isn't relevant there. The entries which are specific to a particular type have already been covered in chapter 5 *Further Types of Data*.

Different types of frame, however, do have many entries in common. A typical frame menu is shown below and the items explained:



Draw

This option is specific to the type of frame.

Save

This allows information about the frame to be saved. There are two options; **Resource** and **Button.** Resource saves just the data associated with the frame, for example the text which it contains, a copy of the draw file attached to it etc. Button saves details of the frame itself and its link information. It also saves a copy of the resource into the **!GenLib** library directory.

Sliding off either option will display a dialogue box for the name of the file to be created and an icon which can be dragged and dropped onto a directory viewer to create the file.

Copy

This makes a copy of a button, ie the frame, its data and its link information are all replicated. The centre of the new frame is placed at the mouse position.

Shared

This is one of the more advanced features of Genesis for saving memory and disc space. Normally, all the data associated with a page is saved in the page's subdirectory and is said to be *local* to that page. However, if the same drawing, Maestro file etc is to appear on several pages, this method of storing the data is wasteful since one copy is required for every page it appears in. An alternative method is to make the data *shared* between all the pages in the application. This means that just one copy is required however many times it appears. Shared data lives in the top level of the application directory rather than in one of the page subdirectories.

Selecting this option toggles the shared state, ie if the data is currently local it is made global and vice versa. A tick by the option indicates that the data is currently global, no tick means that it is local.

Note that this option is not available for text and word frames.

Link to

This option is used for forming links between a frame and a page. Sliding off this option displays a Link tool which can be dragged and dropped on top of the destination page. A tick then appears next to the **Link to** option to indicate that a link has been formed from that frame.

To change the destination of a link, repeat the process for the new destination page. This deletes the old link and replaces it by the new one. To delete the link completely, select the **Link to** option. This breaks the link and the tick will be removed to indicate that the frame is not linked to anything.

Note that this option is not available for any frame which has an implicit action associated with it. For example, Maestro frames cannot be linked to pages since double clicking on them has a predefined action of playing the music.

Border

This option is used for turning the display of the frame's border on and off. Selecting this option toggles the state of the display, ie if the border is currently being displayed, it is turned off and vice versa. A tick by the option indicates that the border is currently being displayed, no tick means that it is not.

Lock

This option is used for *locking* a frame so that it cannot be edited thus preventing accidental changes being made. Selecting this option toggles the lock state, ie if the frame is currently locked, it is unlocked and vice versa. A tick by the option

indicates that the frame is locked, no tick means that it is not.

Delete

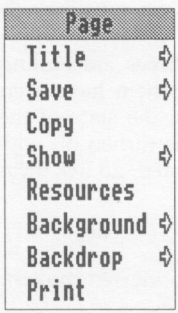
This option deletes the frame and its attached data. If the frame is currently locked, this option is not available.

Page

Sliding off this option causes the Page menu (described below) to appear. This is useful if you are trying to access one of the Page menu options but the background is not visible due to the page containing many frames.

Page menus

All pages have the same page menu, this is displayed below:



The options affect either the page itself or all frames within the page:

Title

This option allows the title of the page to be reset. By default, the first page is called **TitlePage** and subsequent ones **<untitled>.** Sliding off this option produces a writable icon showing the current name. This can be deleted with the delete key and the new name entered.

Save

This allows information about the page to be saved. When a page is edited, the new version is held in the computer's memory. If you want to keep the edits, you have to save the page. An asterisk, '*', after the page name indicates that changes have been made to the page which have not been saved.

There are two options; **Whole Page** and **Script.** Whole page saves any resource files used by the page and the script file describing the page. This can be carried out by selecting the Whole page option or alternatively, by selecting the **Save** option on the Page menu. Script just saves the script file describing the page. In this case, sliding off the option will display a dialogue box for the name of the file to be created and an icon which can be dragged and dropped onto a directory viewer to create the file.

Copy

This option makes an exact copy of a page and displays it. It is a quick way of creating several pages which are similar. Subsequent edits can be made to the pages to change their title, data, links etc without affecting the original.

Show

There are two options; **Links** and **Frames.** These are used for indicating the position of frames and showing which have links from them.

When **Show ▸ Frames** is active, all frames are outlined in red. In the case of **Show ▸ Links,** frames with links from them have a cross drawn through them. Selecting either of these options toggles the state of the display, ie if frames are currently being displayed, the display is turned off and vice versa. A tick by the option indicates that the display is activated, no tick means that it is not.

Resources

This option displays a special directory viewer showing the resources which are local to the page.

Background

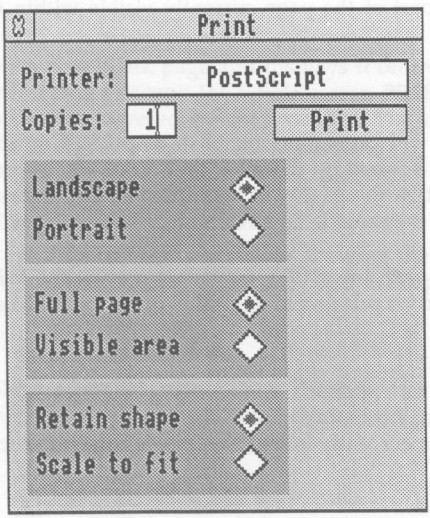
This option allows the background colour of the page to be altered. Sliding off it will give a menu of colours available in the current screen mode. If one of these is selected, any areas of the page not occupied by frames will be displayed in this colour.

Backdrop

This option is only available if the page contains a backdrop. Sliding off this option displays a further menu of the actions available: **Save, Shared** and **Delete.** These act in a similar manner to the corresponding Frame menu options. Save allows a copy of the resource to be saved. Shared toggles the state of the resource between being global to all pages and local to just this one. Delete erases the backdrop.

Print

Selecting this option produces a dialogue box which allows you to choose the type of hard copy you want. Before using this option, the appropriate RISC OS printer driver must have been run so that the format of the output is correct for your particular make of printer. Details of how to set up and use printers are provided in the *Archimedes User Guide* provided with the machine. The dialogue box provides the following options:



Printer:

This option refers to the currently installed RISC OS printer driver. If this contains **None,** then you have not installed a printer driver. You must do so before being able to print.

Copies:

This option allows you to set the number of copies of the page to print. By clicking inside the writable icon it is possible to enter the number of copies required.

Landscape / Portrait

This option allows you to choose the orientation of the printed page on the paper. **Lanscape** printing has the longer side of the paper horizontal, **portrait** has the longer side of the paper vertical.

Full page / Visible area

Selecting **Full page** means that the full contents of the page will be printed even if only a small part is currently visible within the window. Selecting **Visible area** means that only the part of the page currently visible within the window will be printed.

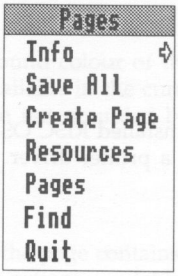
Scale to fit / Retain shape

Selecting **Scale to fit** will cause the output to be printed on the full extent of the paper, stretching or squeezing the text and graphics approriately. Selecting **Retain shape** means that the page will be printed so that the *aspect ratio* of the text and graphics remains the same ie no stretching or squeezing will take place.

When you are happy with the printer options, selecting the **Print** button will start the printing. To interrupt printing at any time, press the ESC key.

Application Menu

The application menu is obtained from the application icon on the menu bar. When an application is currently in use, this menu appears as follows:



Info

Sliding off this option provides information about the application.

Save All

This option saves the current state of all pages within the application. It is similar to selecting the **Save ▸ Whole Page** option from the Page menu for every page which has been edited.

Create Page

This option creates a new blank page and displays it on the screen. The window it is displayed in will initially occupy the whole screen extent apart from the area at the bottom of the screen covered by the icon bar. The title of the page will be **<untitled>** and the background will be set to white.

Resources

This option displays a special *resource* viewer showing the shared resources which the application owns. More details will be provided in chapter 9, *The Genesis Filer*.

Pages

This option displays a special *page* viewer showing the pages contained in the application. More details will be provided in chapter 9, *The Genesis Filer*.

Find

This option is used for searching through the text frames of an application to locate a particular series of characters. Clicking on the option displays the Find text window. This contains a dialogue box for the string to be searched for, a **GO** box for starting the search, a **NEXT** box for recommencing the search at the previously found page and two option boxes and two sets of radio buttons..

If **Case sensitive** is selected, then the case of the letters is taken into account during the search. This means that *Ab* will only find *Ab* and not *AB*, *ab* or *aB*. If the option is not selected then the case is irrelevant and *Ab* will successfully find all four alternatives.

If **Find all** is selected, then the whole application is searched and all the pages found are opened, on top of each other. If not, then the search will stop as soon as the first page is found and the **NEXT** button has to be used to locate subsequent ones.

If the **Use index file** option is selected then the application itself is not searched.

Instead its associated *index file* is used. This speeds up the searching and provides a method of dictating which pages will be found by a given search. Index files are covered more fully in Appendix D, *Index Files*.

If the **To GenFiler** option is selected then the pages that are found are displayed by the GenFiler from where thay can be selected in turn by the user. See chapter 9, *The Genesis Filer* for more information.

Quit

This closes down the Genesis application and removes the icon from the icon bar. If the application has been edited in any way and these changes have not been saved, then a window will appear reminding you of this fact and giving you the option of returning to the application to save them. If you go ahead and quit anyway, these edits will be lost.

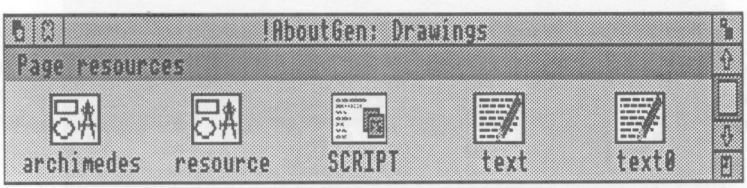
9 THE GENESIS FILER

This chapter covers the actions of the Genesis Filer. This is the part of the Genesis system which is responsible for the general handling of the pages and data files contained within a Genesis application. Its use is fully automated; whenever it is required it will be started up and its icon will appear on the icon bar at the bottom of the screen to indicate that it is running.

Resources

The *resources* of a Genesis application are the data files which it owns. Each resource can be either *local* to a particular page or *shared* amongst all pages. By default, when a data file is attached to a frame, a copy of the file is made which becomes a local resource of the page containing that frame. This means that if you attach the same data to several frames on different pages, several copies of the file will be made, one for each page. This is obviously wasteful on space. In this case, a better solution would be to make the file a shared resource. This would mean that, however many pages used it, the application would only keep one copy.

To demonstrate this, an example is needed. Start up the **!AboutGen** application and display the Drawings page. Then point within the page, press the menu button and select the **Resources** option. This will start up the Filer which will display a special *resource viewer* showing the resources belonging to that page:



These resources consist of two draw files, two text files and a script file which will be explained later. This is what you might have expected since the page contains two draw frames and two text frames.

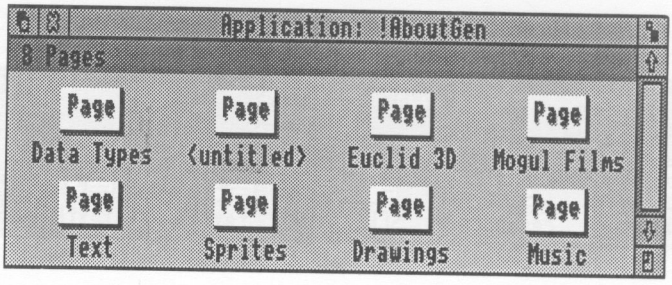
Now, point at the application icon, press the menu button and select the **Resources** option from the application menu which pops up. This displays another resource viewer for the resources which are shared throughout the application. In this case, the resource viewer is empty indicating that there are no shared resources. Again, this is reasonable, because all the pages are about different things.

Close down the two resource viewers, display the Frame menu for one of the draw frames and click on the **Shared** option. This changes the resource from being local to being shared. If you reopen the viewers onto the the page and application resources you will see that the corresponding draw file is displayed as an application resource rather than a page resource.

You can tell whether a resource is shared by pointing at the frame to which it is attached and displaying the Frame menu. A tick by the **Shared** option indicates that the resource is shared, otherwise it is local.

Pages

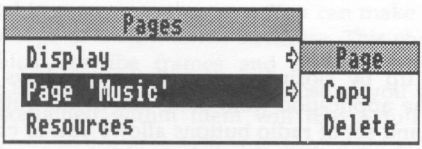
Another action of the Filer can be seen by pointing at the application icon, pressing the menu button and clicking on the **Pages** option. This displays a *page viewer* giving details of the pages held within the application. The top line indicates the total number of pages. Then an icon for each is displayed together with the first 12 characters of its title:



The Filer Page menu

This provides an easy method of manipulating pages. Point at one of these icons, press the menu button and a Pages menu will pop up. The first option, **Display**, is common to all page viewers. It allows the format used for displaying the pages to be changed. You can choose between large icons, small icons or full information.

The second option will be called **Page** followed by the title of the page, for example **Page 'Music'**. Now slide off this option to obtain the submenu:



This contains two options, **Copy** and **Delete:**

Copy provides an alternative method of copying a page.

Delete allows you to delete a whole page. This is the only way to delete a page in Genesis. It was felt that deleting a page is such a drastic step to take that such an option should not be placed where it could easily be selected accidentally and therefore it is only available via **!GenFiler.**

The final option is **Resources.** Selecting this will display a further resource viewer showing the resources which are local to that page.

Opening pages directly

Another feature of the page viewer is that you can double click on any of the icons within it to make that page appear on the screen. Try this now by double clicking on the **Mogul Films** page icon. The page describing Mogul films will be drawn at the front of the screen. This provides a quick method of checking the contents of any page.

Making links directly

The page icons in the Filer can also be used to create links. Previously, you have always created links by dragging the Link tool from a frame menu onto another page. However, with the Filer, you can make a link by dragging a page icon directly onto a frame.

To try this, double click on the **Euclid 3D** page icon to open the page about the Euclid 3D drawings. Create a new frame in the top right-hand corner and type the text **`Film`** into it. Now, drag the **Mogul Films** page icon from the page viewer and drop it on top of the frame. This will create the link between the frame and the

page. You can check that a link exists by pointing within the frame, pressing the menu button and verifying that a tick has been placed beside the **Link to** option. Finally, double click within the frame - the Mogul Films page should appear at the front of the screen.

Finding pages

Before continuing, tidy up by closing down all the windows currently on the screen. Now, point at the application icon, press the menu button and select the **Find** option. The right-hand set of radio buttons allows you to choose between the options **Open page** and **To GenFiler**. By default, the former of these is selected. This means that each page found to contain the text will be opened. If, however, you select **To GenFiler** the pages will not be opened. Instead the found pages will be displayed as icons in a page viewer.

To try this, enter the word **text** into the dialogue box, select **To Genfiler** and **Find all** and click on **GO.** A page viewer will appear containing just the icons of the pages which contain the word **text.** Check this by opening each page up in turn by double clicking on its icon.

Note that the buttons will often be used in the combinations **Find first** and **Open page** or **Find all** and **To Genfiler**. When you are opening pages, it is usually preferable to have them identified one at a time so you have enough time to view each before the next one appears. However, if you are just sending the pages found to the Filer then it is better to find them all in one go so that they all appear in the same page viewer. You can then view them independently by double clicking on them from the page viewer.

10 SCRIPT FILES

Each page of a Genesis application is described in a Script file. This contains all the details about the individual frames and general page features such as the backdrop which are required to reproduce the page. You can make use of these files without any special knowledge of their internal structure. This chapter looks at the uses of *button scripts* which describe frames and *shared scripts* which describe pages without detailing the contents of the files. Those of you who are interested in the script language contained within them will find a full description in the final chapter.

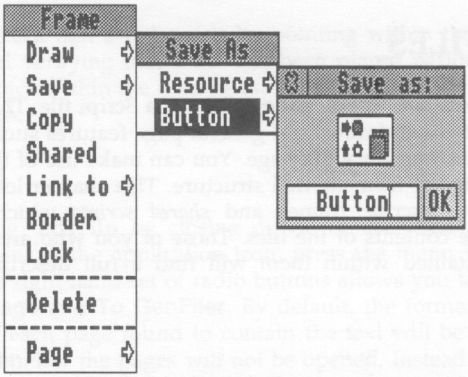
Button scripts

The script file describing a page can be subdivided into the part of the script which defines the general page attributes plus separable blocks of script for each of the frames. These frame blocks contain details of the size of the frame, its link information and the name of the resource which is attached to the frame. They are referred to as **button scripts**.

The **Save option** on Frame menus provides two alternatives; you can save either the resource attached to the frame or its button script. Using the latter, you can copy frames from one page to another. So far, you have seen how individual frames can be copied within a page using the Frame menu **Copy** option. In addition, you have used different techniques for copying the contents of whole pages. However, this is the first time that the possibility of copying individual frames between pages has been mentioned.

To try this out, start up the **!AboutGen** example application and display the **Drawings** page. Then create a new page by pointing at the application icon, pressing the menu button and selecting the **Create Page** option. Resize and reposition this page so that it occupies just the top right-hand quarter of the screen and the majority of the **Drawings** page is visible behind it.

The aim is to copy the frame containing the picture of the Archimedes onto the new page. To do this, point at the Archimedes, press the menu button and slide off first the **Save** option and then the **Button** option. This will display a button icon and dialogue box:



The dialogue box is provided so that you can name the button file when you are saving it in a library. However, in this example the name is irrelevant therefore you need not change it from its default of **Button.** Point at the button icon, press and hold down the left mouse button and drag the icon onto the new page. When you let go of the button, a copy of the frame will appear in the new page, centred at the mouse position.

Note that the size of the frame is exactly the same in the two pages showing that the size and details of the contents of the frame have been copied. In fact, all the details about the frame are the same. For example, because the border was turned off in the **Drawings** page, it is also turned off in the new page. Similarly, if the original frame had possessed a link to another page, the copy would also have been linked to the same destination.

What happens to the non-text resources

The button script contains the definition of the frame including any link information and in addition the name of the resource attached to the frame. However, saving a button script has an affect on the resource itself. This resource is copied into **!Genlib,** the Genesis library.

You have already encountered two types of resource; local and shared. In addition, a third type called library resources exist. These are resources which live in the directory !GenLib. When a button is placed on a page, a check is made to see if its named resource exists as follows:

1. As a local resource of the page

2. As a shared resource of the application

3. As a library resource in !Genlib

If a local resource of the given name exists then it is used. If not then any matching shared resource is used instead. Failing that the library is searched. If the named resource exists there then it is copied from the library into the application directory as a shared resource. If the resource does not exist at all then the frame is left blank.

When you copied the button script above, two actions took place:

1. Extracting the button script caused a copy of the draw file to be saved in !Genlib.

2. Placing the button on the page caused Genesis to try to locate the draw file. Since it was available neither locally nor as a shared resource, it was copied from !Genlib into the application as a shared resource.

You can check this by pointing at the frame in the new page and pressing the menu button. The Shared option will have a tick beside it.

What happens to text resources

One important consequence of this is that text will not be copied. Text files cannot be shared - they can only be local to individual pages. This is a necessary restriction due to the fact that the contents of text frames can be edited directly within Genesis. Displaying and editing the same file in two pages at the same time would lead to confusion!

Try copying the text frame containing the word **Drawings** onto the new page to see what happens. The frame will be copied along with all its information about font style and colour but no text will appear.

Predefined buttons

As supplied, !Genlib contains a number of useful predefined buttons. You have already met one of these in the chapter on databases. Each of these buttons requires a resource file which will be used to represent it. These resources are typically sprites or draw files and are also supplied within !GenLib.

The actions of some of these buttons are as follows:

Another Makes a copy of the current page, creates a link between itself and the new page and turns itself into a Next button. Any non-text resources belonging to locked frames are made *shared* so that they appear on the new page. Text frames and unlocked frames will remain blank on the new page.

Next Displays the page created immediately after the current one

Previous Displays the page created immediately before the current one

Last Displays the last page to be created.

Home Displays the title page.

Try placing them onto your new page and examining their actions.

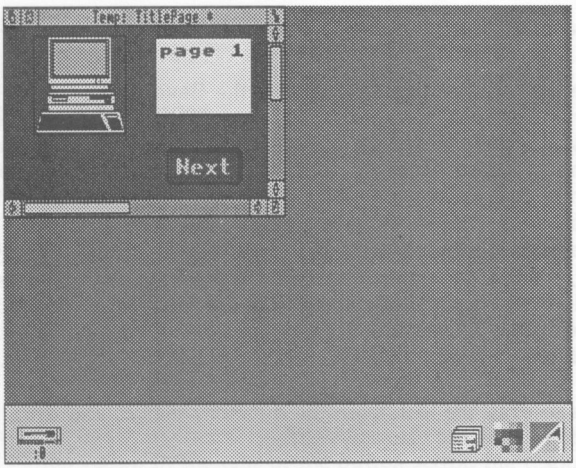
Shared scripts

In the above section, you saw how button scripts can be used. This section moves on to look at the script files produced for entire pages.

Preparation

For this, you are going to require a new application. Close down the current one by selecting the **Quit** option on the application menu. Then start up the Genesis Editor and create a new application. This is only going to be required for the duration of this chapter so a name such as **!Temp** to remind you to delete it later would be a good choice.

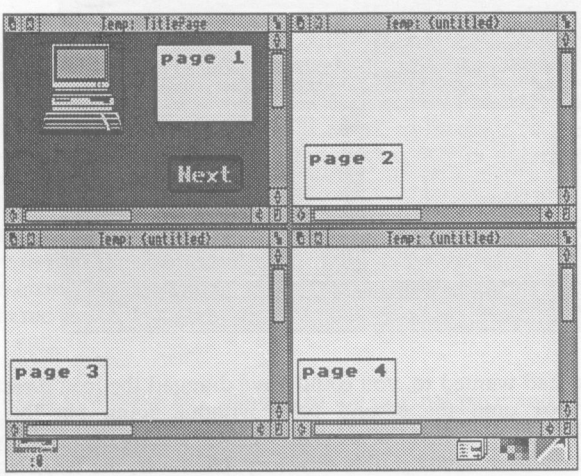
Reduce the size of the first page so that it occupies just the top left-hand corner of the screen. Then, create two frames in the top half of the page. Into the first, type some text such as **page 1**. Into the second, drop a draw file such as the Archimedes drawing which lives in the **Draw** subdirectory of !Genlib. Drag a copy of the Next button from the Button subdirectory of !Genlib, onto the bottom right-hand corner of the page and resize it so that it occupies less space. Finally, use the **Background** option of the Page menu to change the page background colour to red:



Now, use the **Create Page** option on the Application menu to create a second blank page. Reduce the size of this page to again be roughly half the screen width and height and move it to the top right-hand corner of the screen so that the two pages are visible side-by-side. In this page, create one frame in the bottom left corner and type the text **page 2** into it.

Now, make two copies of this page using the **Copy** option on the Page menu and place them in the bottom corners of the screen. Alter the text of their frames to read **page 3** and **page 4** respectively.

Your screen should now look roughly like the following:



Using the Save Script option

You saw in the first part of this chapter that the **Save** option on a Frame menu gives two alternatives, one of these being to save the button script file for the frame. Similarly, the **Save** option on a Page menu also gives two alternatives, the second of which, **Script,** saves the *shared script* ie the script file for the page.

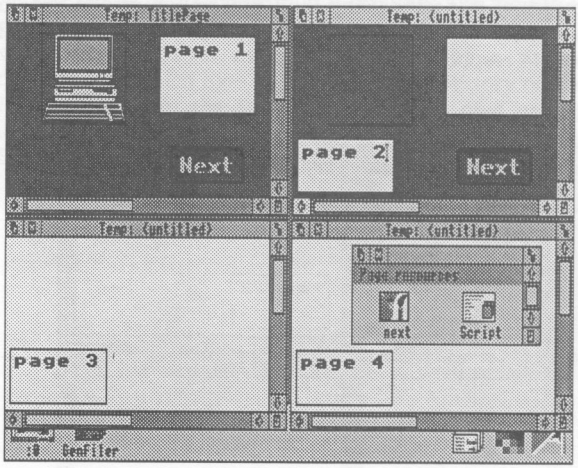
To see how this works, point at the application menu, press the menu button and select the **Resources** option to display a resource viewer containing the shared resources. Next, point at an empty area of the top left-hand page, press the menu button and slide off both the **Save** and **Script** options. This will display an icon and dialogue box. Drag the icon and drop it on top of the resource viewer. This saves the shared script into the shared resources area of the application.

The next step is to drag the icon from the directory viewer and drop it on top of the right-hand page. This has two main effects:

1. The frames which are described in the file are added to those already existing in the page

2. Any page characteristics described in the file replace those of the page

As a result, the second page gains two frames and its background changes to red:



Note that if you had wanted to, you could have dragged the icon directly from the menu display box onto the second page without going via the application directory. This would have had the same affect since a copy of the script file would automatically have been saved in this directory. However, the two stage approach

means that all the actions are clearly visible and so helps to explain what is going on.

What happens to resources

The resources of the buttons described in the shared script are neither copied to !Genlib nor made shared. Since the second page does not contain any local resources with the same names as those used by the first, both the draw frame and the text frame are empty.

If you want the picture to appear in the second page then you can achieve this by pointing at the frame in the original page, pressing the menu button and selecting the **Shared** option. The next time that you open the second page, the Archimedes will appear.

The affect of extracting a shared script therefore contrasts with the affect of extracting a button script. In the case of the button script, the resource is added to the library thus making it available to all pages. However, in the case of the shared script nothing happens to the resource. This difference gives you greater flexibility in the way in which you construct pages.

The basic action of using a shared script is to add some frame definitions without affecting resources. Thus, shared scripts can be used to copy the structure of a page rather than its contents. You can choose exactly which resources are made available by deliberately making the original resource shared. If all are local then the page gains just the frame layouts, if on the otherhand all are shared then the page gains a copy of the entire page.

Editing the scripted page

There is another big difference between what happens when a shared script as opposed to a button script is added to an existing page.

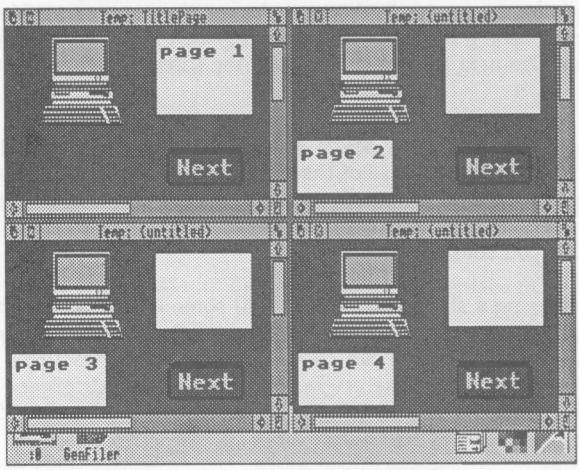
In the case of the button script, the script is added to the page's script. Therefore it becomes a normal part of the page and there is no way of telling that the button was added in this manner.

With a shared script an extra *note* is placed within the page's script file telling Genesis to try to find the shared script file and use its contents whenever the page is opened. To enable this to happen, the shared script file has to be stored just like any other shared resource. Thus, it can then be seen by all the pages in the application.

This means that if you make any changes to the original page and then re-save the shared script from it these changes will be reflected in all the pages which rely upon the shared script file.

To see this, copy the shared script from the resource viewer into the last two pages as well. Now return to the original page, turn off the border for the draw frame and resave the page's shared script by dragging it onto the resource viewer again. Now save the application in this new state using the **Save All** option from the Application menu. Close down all but the original page and finally open them all up again by double clicking on the Next buttons of each page in turn.

As each page appears, you will see that they all have the draw frame borders turned off. Hence only the original page had to be altered in order to make the change to all four pages:



Note that in this case, you are strongly advised to resave the shared script by dragging it onto the directory viewer as described rather than by dragging it directly onto the second page. The latter approach would mean that the copy of the shared script file in the application directory would be correct since, as was stated above, a copy would automatically be made there overwriting the existing copy of the same name. However, the act of dropping it onto the page would mean that an attempt would be made to include another reference to the shared script in the page script.

Local variations

The final question which needs answering is *What happens if one of the frames from the shared script is edited in a page which includes it?*. To find out the answer, try moving and re-sizing the draw frame in the last page. Close the page, choosing to save the edits made when prompted and then open it up again by double clicking on the Next button of the previous page. You should find that your edits have remained in force - the local changes which you made have been remembered and used.

However, now try deleting the frame and seeing what happens. This time, when you next display the page, the frame will have reappeared in its original form.

Therefore, you can effectively make a *local variation* of any of the included frames which will take precedence over the included shared version. However, if you try to delete the frame, you will return to using the shared version again!

Applying this to databases

A good use for shared scripts would be in designing a card index type database. You could design the layout of the record cards on a page and then save the definition as a shared script file. You would then create a second page and cause it to adopt the same structure as the original simply by dragging the shared script file onto it. All the rest of the record cards for the database could then be created by copying this second page or through use of the Another button.

This method may sound a little complicated but it has one tremendous advantage over simply copying the first page many times. If at a later date you choose to add a new field to the record card or to change the layout in some way you only have to make the change once! To do this, make the change to the first record card and then re-save the shared script file, overwriting the previous version of it. When you next open any of the other record cards they will have inherited the changes automatically.

Using the shared script file does not prevent you from making minor alterations to a few copies of the page when necessary. For example, if one of the frames proves to be too small for just one record and you don't want to change its size throughout all the pages then you can just make the edit to the single page which needs it. In addition, you are free to add extra frames to any of the copied pages if you want to include miscellaneous items of data in some cases. These frames will belong just to the page they are added to and will not affect the included shared script.

11 GENESIS SCRIPT LANGUAGE

This chapter takes a look at the contents of the script files produced by Genesis. It explains all the entries of such a file and provides you with the information required to either edit existing script files or create new ones of your own if you so wish.

The script language which Genesis uses is called *GenScript*. This has two general rules which apply throughout. The first is that all keywords must be given in upper case. The second states that any textual data for arguments should be enclosed within quotation marks. For example:

```
ACTION FIND "London"
```

Page Definitions

Every page has a separate script file which defines all the attributes of the page. When you create a new page its script file will be as follows:

```
DEF PAGE "<untitled>"
DEF WINDOW 2 180 1238 980 0 0
    BACKBOX ON
    TITLE ON
    QUITBOX ON
    FULLSIZE ON
    VSCROLL ON
    SIZEICON ON
    HSCROLL ON
    WORKAREA 1236 936
END WINDOW
BGCOL 0
END PAGE
```

The definition of the entire page is enclosed between a pair of statements; **DEF PAGE** and **END PAGE**. It contains information about the following features of the page:

Title

The **DEF PAGE** statement is followed by the title of the page, in this case it is **<untitled>**.

Window definition

Next comes the definition of the window in which the page is displayed. This is similarly enclosed by the pair of statements **DEF WINDOW** and **END WINDOW**. The definition starts by stating the position and size of the window. The six numbers required appear in the following order:

minimum-x minimum-y maximum-x maximum-y scroll-x scroll-y

The following seven lines define the states of the window flags. By default, all seven of the window flags are ON. This means that the window will be fully scrollable, moveable and resizeable. You can elect to remove any of the window features by changing the appropriate line of the window definition. Simply replace the ON by OFF.

The final part of the window definition is the **WORKAREA** statement. This defines the maximum width and height of the window. The backdrop, if one exists, will be scaled to fit this area.

Background colour

The background colour to be used is defined by the **BGCOL 0** statement.

Adding backdrops

If you added a backdrop to the page then the definition for this would also appear in the script file. For example if the page had a backdrop consisting of a draw file called a3000 the script file would change to the following:

```
DEF PAGE "<untitled>"
DEF WINDOW 2 180 1238 980 0 0
      BACKBOX ON
      TITLE ON
      QUITBOX ON
      FULLSIZE ON
      VSCROLL ON
      SIZEICON ON
      HSCROLL ON
      WORKAREA 1236 936
END WINDOW
```

```
DEF BACKDROP
DRAW a3000
END BACKDROP

BGCOL 0
END PAGE
```

Furthermore, if the Backdrop had been locked to prevent accidental deletion then the word **LOCKED** would be included in the script:

```
DEF BACKDROP
DRAW a3000
LOCKED
END BACKDROP
```

Frame Defintions

All the frames on a page are defined within the body of the page definition. That is they come one after another following the **BGCOL** line and before the **END PAGE** line of the page definition.

A simple text frame definition might look like this:

```
DEF FRAME "MyName"
TEXT "text"
STYLE "Trinity.Medium"
SIZE 320
FGCOL 7
BGCOL 0
AT 18 -148 312 -16
END FRAME
```

General syntax

In general the syntax for a frame definition is:

```
DEF FRAME <name>
<type> "resourcename"
<resource specific information>
<position>
<optional ACTION> <arguments>
<optional LOCKED>
<optional BORDER>
END FRAME
```

The entire definition is again enclosed between a pair of statements; **DEF FRAME** and **END FRAME**. The last three items in a general frame definition are all optional. However, if they are given, they must be in the order shown above. These and the other parts of the definition are defined below:

<name>

This is a string enclosed within double quotes giving the name of the frame. The name is created by Genesis for its own use to enable it to recognise the different frames. When you create a frame using the mouse, Genesis will make up a name automatically based on the time and date together with a sequence number. This should ensure that no two frames on the same page have the same name!

Frames on different pages may have the same name without causing any difficulties. Indeed they quite often will have the same name if pages are copied or if the Another button is used. This is expected to be a useful feature in the future since it will allow advanced applications to do interesting things such as restricting the search of a Genesis database to certain fields. However, at the time of writing, frame names are of little actual use.

<type>

This can be any of the standard data types supported by Genesis:

TEXT	Text frames
WORD	Word frames
DRAW	Object orientated graphics
SPRITE	Bit mapped graphics
EUCLID	3D graphics
FILM	Animated graphics
MUSIC	Maestro music

Alternatively, it can be a general RISC OS file type. These take the form:

FILE_FFB	A BASIC Program

where the last three characters are the RISC OS file type code.

Finally there are three other types which are recognised. These are

APPLICATION	A RISC OS application
DIRECTORY	A directory
UNTYPED	A non-date stamped program eg TWIN

<resource specific information>

This depends upon which of these types the frame is. Details are given for each type in the following section.

<position>

This specifies the location of the frame on the page. It consists of an **AT** statement followed by the position of the bottom left and top right corner of the frame relative to the page origin. The page origin is the top left corner of the page and the coordinates run from left to right for positive x and from bottom to top for positive y. This means that for a frame to appear on the page, both y values must be negative with the first having a larger magnitude than the second.

<optional ACTION>

This may be used to define an explicit action for a frame such as a link to another page. The action definition starts with the word **ACTION** and is followed by an action statement. It is defined more fully in a later section.

<optional LOCKED>

The word **LOCKED** may appear within the frame definition indicating that the frame is locked. If it is not present the frame is assumed to be unlocked and therefore editable.

<optional BORDER>

Similarly, the word **BORDER** may appear within the frame definition indicating that a border is to be drawn.

The resource definition

Part of a frame definition consists of the resource type followed by information which is specific to this type. Each of the types allowed are described below:

Text frames

For a text frame, the style and size of the character font should be supplied followed by the foreground and background colours which are to be used.

The size given in the script file is in units of 1/16th of a point. Thus 320 corresponds to 20-Point text. If the height of the characters is not given explicitly then it is assumed to be equivalent to the character size.

The following is another definition of a text frame which specifies that the system font is to be used. Note that the script contains a line setting the size to zero. Zero has a special meaning in this situation, it instructs Genesis to use the default system font size. This feature is useful since it results in a greatly increased speed of redrawing than is possible with the scaled characters.

```
DEF FRAME "Wed,08 Nov 1989.10:34:57;2"
TEXT "text1"
STYLE "System Font"
SIZE 0
FGCOL 7
BGCOL 0
AT 344 -764 476 -704
END FRAME
```

Word frames

In the following example of a word frame, the text used for the word can be seen enclosed in quotes on the second line:

```
DEF FRAME "Thu,14 Sep 1989.14:25:30;1"
WORD " "
FGCOL 7
AT 318 -292 462 -24
END FRAME
```

Note that in this example, the background colour is not stated. This means that the frame is to be transparent. Since the contents is just a space and there is no border this frame will be invisible.

Sprite frames

All that is required for a sprite frame is the name of the file containing the sprite and the position at which the frame is to be drawn:

```
DEF FRAME "s1"
SPRITE "Sprite1"
AT 24 -208 354 -16
END FRAME
```

Draw frames

Similarly for draw frames:

```
DEF FRAME "d1"
DRAW "draw1"
AT 24 -208 354 -16
END FRAME
```

Euclid frames

These are rather more complicated. A typical Euclid frame appears below:

```
DEF FRAME "e1"
EUCLID "castle"
STYLE 2
SCALE 1
XPOS 0
YPOS 0
XROT 0
YROT 0
AT 36 -792 462 -416
END FRAME
```

The **STYLE** value determines the display mode and light settings as defined on the Info window:

Faces only	0
Wireframe	1
Faces and edges	2
No lights	0
Lights on	512
Front light	1536

The value represents the total of the options. For example, faces and edges together with lights on produces a value of 2 + 512 ie 514.

The other values represent the zoom, translation and rotation achieved using the Euclid panel. The figures given in the example are the default values obtained when no editing has taken place.

Mogul film frames

These are similar to draw and sprite frames requiring just the file name and a position:

```
DEF FRAME "m1"
FILM "litrings"
AT 724 -408 1180 -104
END FRAME
```

Maestro music frames

Again, these take the standard form:

```
DEF FRAME "mus3"
MUSIC "Entertain"
AT 518 -520 642 -384
END FRAME
```

Other types of frame

The type of other files takes the form **FILE_** followed by the three digits which represent the standard RISCOS filetype. For example, a frame which holds a BASIC program would appear as follows:

```
DEF FRAME "bas1"
FILE_FFB "basprog"
AT 404 -364 582 -232
END FRAME
```

There are three sorts of object which can be attached to a frame in an entirely different way. In each case, instead of a copy of the object being stored with the page as a resource, the frame definition merely remembers the full filing system path name of the object. These three types of object are; directories, applications and non-date-stamped files.

A typical directory frame would be defined as follows:

```
DEF FRAME "dir"
DIRECTORY "adfs::win.$.app"
AT 120 -316 256 -204
END FRAME
```

In this case, double clicking on the frame will cause Genesis to ask the Desktop filer to open a directory viewer of the directory **adfs::win.$.app**.

In the case of an application, the frame definition will be of the following form:

```
DEF FRAME "mag"
APPLICATION "adfs::win.$.app.!magnifier"
AT 18 -168 132 -52
END FRAME
```

Again, double clicking causes Genesis to use the full path name as given to start up a new application task. In this case, the magnifier utility will be run. Note that Genesis searches various directories in the current filing system before resorting to using the full path name in order to avoid unnecessarily provoking a **Please insert disc** message.

Files which have no file type associated with them are called `UNTYPED`. An example of this sort of file is the TWIN program editor:

```
DEF FRAME "twin"
UNTYPED "adfs::win.$.library.twin"
AT 306 -176 730 -16
END FRAME
```

In this case, the full path name is used directly to run the file.

The action definition

Missing from all the example frames above has been any form of explicit action definition such as a link to another page. Only certain types of frame can have actions, these are:

Text

Word

Sprite

Draw

Euclid

Film

The action definition comes immediately after the position **AT** definition. An example of a frame with an action is as follows:

```
DEF FRAME "s1"
SPRITE "Sprite1"
AT 24 -208 354 -16
ACTION GOTO "0"
END FRAME
```

Genesis supports a variety of other actions such as those provided by some of the special buttons in !GenLib. At the time of writing, the following special actions are available:

ANOTHER

NEXT

PREVIOUS

LAST

HOME

COPY

COPYLINK

FIND <arg>

OSCLI <arg>

WIMPTASK <arg>

The first group are simple one word actions and require no arguments. The members of the second group each accept a single argument. The GOTO described above in the section on Links falls into this second category. In detail, the actions supported are as follows:

GOTO

The simplest action statement is a **GOTO** used to represent a link to another page. For example:

```
ACTION GOTO "3"
```

This will result in a double click on the frame opening page **"3"** of the application. The title page of the application is always page **"0"** and therefore **GOTO "0"** can be used to open the title page.

It is also possible to make a link to a page of a separate Genesis application. This is also represented in the script file by a **'GOTO'** . For example:

```
ACTION GOTO "adfs::$.!DataBase.5"
```

Double clicking on a frame containing this definition would cause the Genesis application !DataBase to be run and page "5" of it to be displayed.

ANOTHER

This creates a copy of the page and makes a link from itself to the new page. It then changes its action to a **GOTO** directed at the new page and changes its resource from **Another** to **Next.** The resources of the page are not copied unless the frames to which they are attached are locked.

NEXT

This opens the next page in the application. Thus, if used on page "4", it will cause page "5" to be opened when the frame is double clicked on.

PREVIOUS

This opens the previous page in the application. If used on page "4", it will cause page "3" to be opened when the frame is double clicked on.

LAST

This opens the last page of the application.

HOME

This opens the first page of the application, ie it is equivalent to **GOTO "0"**.

COPY

This copies an entire page including all its local resources.

COPYLINK

This creates a copy of the page and makes a link from itself to the new page. It then changes its action to a **GOTO** directed at the new page and changes its resource from **Copylink** to **Next.** The local resources of the page are all copied.

FIND

This takes a textual argument and searches the application looking for the given text. Thus, **FIND "London"** causes Genesis to search for pages containing the text London. It obeys the options which are currently selected in the Find window.

OSCLI

This takes a textual argument and passes it to the Operating System Command Line Interpreter. This is an extremely powerful facility since it allows you to build buttons with complex actions. For example, using the Dabs Press ABC BASIC compiler, it is possible to write a BASIC program which can be run as the result of a * command. This program can therefore be run as the result of double clicking on a Genesis frame. This is used in the tone dialing application, **!Phone.**

WIMPTASK

This is very similar to OSCLI except that the command is passed to the Window manager as a request to start a new separate task running.

Built in functions

In all **ACTION** statements where an argument is taken, it can be a quoted string such as "CAT". Alternatively, you can use one of the built in functions to extract data from the frame. At the time of writing, the following functions are available:

RESOURCENAME
RESOURCETEXT

The **RESOURCENAME** function returns the textual name of the resource which is attached to the frame. In the case of a word frame, it returns the text as displayed by the word frame since there is no separate resource.

The **RESOURCETEXT** function can only be applied within text frames and returns the first line of text from the resource file. For example, you might set up a frame whose action was:

```
ACTION FIND RESOURCETEXT
```

Double clicking on this frame would initiate a search for the text currently held in the first line of the frame. Note that the Find would often find this page itself as the first and therefore if the Find first option is selected in the Find window nothing obvious may happen.

INCLUDE files

This is the mechanism by which shared script files are implemented. For example a page definition from a simple card index might look like this:

```
DEF PAGE "Record 10"
DEF WINDOW 2 180 1238 980 0 0
      BACKBOX ON
      TITLE ON
      QUITBOX ON
      FULLSIZE ON
      VSCROLL ON
      SIZEICON ON
      HSCROLL ON
      WORKAREA 1236 936
END WINDOW

INCLUDE "RecordForm"
END PAGE
```

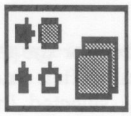
Here all the information about the frames and resources for the page comes from the shared script file **RecordForm.** This must exist within the application's shared resource area. It will be read in and processed whenever the page is opened. This script will be dealt with exactly as if it were inserted at the point where the `INCLUDE` statement occurs.

Any frame definitions coming after the `INCLUDE` which happen to have the same frame name as a frame defined by the included file will take precedence. This allows local variations to be made.

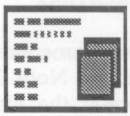
It is possible to include many script files to build up very complex structures.

Button scripts

A second type of script file exists. This is known as a button script and is indicated by a different file type and file icon.



Genesis Button file icon



Genesis Script file icon

Rather than defining a whole page a button script usually defines just a single frame. For example the ANOTHER button provided in !GenLib. When one of these button script files is dragged onto a page the script which it contains is added to the script of the page. Thus any definitions which it contains become part of the structure of the page.

An important difference between a button script and a full page script is that the frame definitions within button scripts do not have a full position indication. Instead they use the **SIZE** keyword to give a width and height for the frame. This means that they take their position on the page from the mouse pointer when dropped onto a page. For example:

```
DEF FRAME "Another"
DRAW "another"
SIZE 150 70
ACTION ANOTHER
END FRAME
```

It is also possible to define multiple frames within a button script. However, the use of the **SIZE** keyword to position them would result in them ending up on top of each other. Two further keywords are provided to help minimise this problem. These are **LEFT** and **RIGHT**.

The following example illustrates how they can be used:

```
DEF FRAME "Previous"
SPRITE "previous"
LEFT 200 100
ACTION PREVIOUS
END FRAME
```

```
DEF FRAME "Next"
SPRITE "next"
RIGHT 200 100
ACTION NEXT
END FRAME
```

This button script defines two frames. One for moving on and one for moving back through the pages.

When dropped onto a page the *previous* frame will appear to the left of the mouse pointer and the *next* frame will appear to the right. The separation is governed by the size given. In this case both frames will be 200 by 100 graphics units. The gap between them will be 2*200 = 400 units wide. This gives a relatively neat format but if it is unsuitable the frames can be moved around in the usual way. The separation of 400 units is chosen so that a third frame of the same size could be added between the two. For example:

```
DEF FRAME "home"
SPRITE "home"
SIZE 200 100
ACTION HOME
END FRAME
```

Summary

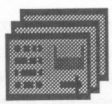
The above sections describe the entire syntax supported by GenScript. Therefore, if you have managed to reach here, you should in theory be able to create new pages of a Genesis application by writing the script files for them. However, it is advised that you start by trying something simpler like editing an existing script file or creating a small button script. If you want to see more examples of the features in use, then all you have to do is to create pages in the normal way and then examine the script file which accompanies them.

Appendix A: CUSTOMISING A GENESIS APPLICATION

There are two ways in which you can modify a Genesis application to give it the appearance of a fully custom-built program.

Application icon

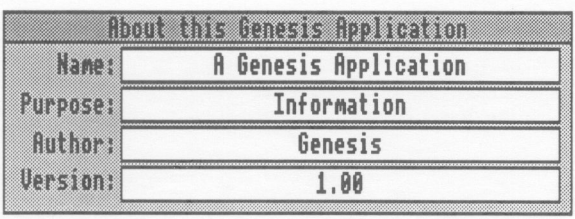
It is a feature of any first-class RISC OS application that it provides an icon to be used to represent itself within the Desktop. This icon should be designed to give a visual clue as to the purpose of the application. By default, all Genesis applications are created with the same icon:



It is straightforward to edit this icon using !Paint in order to give a more meaningful representation of your application. To do this, double click on your application whilst holding down the SHIFT key to open a directory viewer and drag the file !Sprites into the painting program. When you have finished designing your new icon, save the file back in the same place. Note that the sprite name should be kept unchanged.

Information box

All RISC OS applications should also provide an information box from the first entry in the menu which comes from the icon which they place on the icon bar. Genesis provides a default information box for every new application. This takes the form:



You are of course free to change this if you wish. To do so, you will require the Acorn Template Editor program *!FormEd*. The only assumption made by Genesis is that the information box is called **infobox** and that it is contained in a template file called **!Templates** held within the application directory.

Appendix B: LINKING ACROSS APPLICATIONS

It is possible to form a simple link from any frame on any page of a Genesis application to any other page within the same application. Such simple links within the same application present no difficulties when applications are copied or moved around the directory structure of a disc.

However, it is also possible to make a link between pages of different Genesis applications. This can be done in exactly the same ways as making a simple link. That is, either by dragging the link tool from the source frame to the destination page or by dragging a page icon from the filer and dropping it on a frame. When such an inter-application link is used, the effect will be to run the new application and have it open the appropriate page.

The linking information is recorded by means of a full path name reference to the destination application and therefore the link is dependent on the precise disc directory structure. It is probably, therefore, best to use these inter-application links sparingly as they will reduce the portability of your software.

Appendix C: STRUCTURE OF A GENESIS APPLICATION

Each Genesis application is stored in a separate application directory. In the top level of this directory you will always find the following:

!Boot The boot file which is run when the Desktop filer first sees the application directory.

!Run The file which is called to start up the application when the application directory icon is double clicked on.

!Sprites A sprite file containing the definition of the application directory icon.

!Templates A template file which contains the description of the window displayed when sliding off the Info option of the application menu.

In addition other files may exist here. These are the files which are shared by the whole application such as shared resource files or shared script files. There may also be a file called **!Index** which contains an indexed list of keywords for the application.

The application directory will also contain subdirectories representing the pages. The first ten pages use the subdirectory names **0** to **9.** Contained within each subdirectory are all the local page resources and the script file which describes the page.

If the application contains more than 10 pages, another level of directory structure is used. This is to avoid the limit imposed by the Archimedes filing systems on the number of objects allowed within a single directory. On ADFS this limit is 77 which is clearly too small for a large Genesis application.

The next ten pages are located within a subdirectory **b** as directories **0** to **9.** In fact, there is a general method of converting from a page number to a textual directory name by which the page can be found within the application directory. Each digit of the number other than the units digit is converted according to the following table:

```
0 1 2 3 4 5 6 7 8 9
a b c d e f g h i j
```

Then dots are inserted between all the characters of the resulting string. Thus, page **321** is repesented by the directory **d.c.1** within the application directory.

Similarly, the transformation can be applied the otherway around. For example, the page held as **h.a.e.5** has the page number **7045.**

This numbering scheme has the advantage that it allows for more pages to be created than will ever fit on any disc!

Appendix D: INDEX FILES

The find command supports the use of index files. Each application can have an index file containing as many keywords as you like. This index file should be saved as **!Index** in the application directory to which it belongs.

The main advantage of using an index file is that it gives very much faster access to pages than if the exhaustive search method is used. Rather than having to look at each page, and within each page each resource file, the use of an index cuts the time spent accessing the disc to an absolute minimum.

A second advantage of index files is that you can choose which pages are indexed under which keywords. This means that you can ensure the relevance of the pages which are found under a given topic.

Index files can be created directly using !Edit. They are just text files where each line starts with the keyword which is followed by a list of page numbers separated by spaces. For example:

```
England 7
London 7 9
France 8
```

Once you have created an index file and placed it in the application directly under the file name *!Index*, open the **Find text** window and select the **Use index file** option. You can then carry out the search in the usual way.

Note that the index does not have to be in alphabetical order and it does not matter if the same word appears more than once.

Appendix E: INTERNATIONAL VERSIONS

As supplied, Genesis provides all its menu entries, dialogue boxes and other messages in English. However, it can easily be reconfigured to operate in a foreign language. Translated messages are supplied for five other languages:

Country	Code
German	D
Dutch	NL
French	F
Italian	I
Spanish	ESP
Britain	GB

To switch to a different language, SHIFT-D-CLICK on !Genesis and then double click on the **country** file. This will contain **GB** which is the international vehicle identification code for Great Britain. Change this to the appropriate code and resave the file. Next time you run Genesis, the messages will appear in the language which you selected.

Further language translations may become available. Those interested should contact Software Solutions for advice.

INDEX

NB **bold** *entries in the index refer to menu and dialogue box options.*

ACTION 73, 78
Adjust size icon 8
Animation 29
ANOTHER 78, 79
Another button 34-35, 61
APPLICATION 73, 77
Application
    Creation 7, 31
    Icon 17, 85
    Menu 11, 52-54
    Name 7
    Running 15, 17, 40
    Saving 15
AT 71, 73

BACKBOX 69
BACKDROP 71
Backdrop 17, 45-46
Backdrop 50
Background 17, 45-46
    Transparent 19, 25, 30
    Coloured 11, 25, 30, 50
Background 11, 19, 25, 30, 50
Backups 3-5
BGCOL 69-70, 71, 74
BORDER 73
Border 13, 48
Browser 39-41, 48
Button 34, 45
    Action 73, 78
    Predefined 34, 61-62
    Script 47, 59-62, 81-83
Button 47, 59

Caret 10, 24
Case sensitive 37, 53
Clicking
    Double 9, 14
    Left-hand 7-9, 14
    Right-hand 7, 14
Close icon 23
Closing windows 23
Copies 51
COPY 78, 80
Copy 32, 33, 47, 50, 57
Copying
    Frames 32, 47, 59
    Pages 33, 50, 57, 64
COPYLINK 78, 80
Create Page 11, 53
Creating
    Applications 7, 31
    Frames 9
    Pages 7, 11, 14, 32-34, 53
Cursor 10, 30

Database 1, 31-37, 67
DEF 69, 70, 71
Delete 33, 49
Deleting
    Frames 33, 49
    Pages 57
DIRECTORY 73, 77
Directory viewer 4, 12

Dragging 4, 7, 12, 14, 34, 44
DRAW 72, 75
Draw 25, 47
Draw files 12, 25

Edit 89
Editing
    Text 10, 24
    Frames 9
    Resources 25
Editor menu 7
END 69, 70, 71
EUCLID 72, 75
Euclid 27
Euclid 3D pictures
    Panning 28
    Rotating 28
    Zooming 28

Faces only 27
Faces & Edges 27
FGCOL 71, 74
FILE 73, 76
Filer 55
FILM 72, 76
Film 29
FIND 78, 80
Find 36, 53, 58, 89
Find all 36, 53, 58
Find first 58
FRAME 71
Frame 1, 44
    Action 22
    Border 13, 48
    Copying 32, 47, 59
    Creation 9
    Deletion 33, 49
    Editing 9
    Linking from 13-15, 21, 45, 57, 87
    Locating 18
    Locking 41, 48
    Menu 11, 23, 47-49
    Moving 9
    Resizing 9, 24
    Saving 47
    Sharing 48
Frames 18, 50
Front light 28
FULLSIZE 69

GenScript 69-83
GOTO 79-80

Hard copies 51-52
Height 11, 24
HOME 78, 79
Home button 62
HSCROLL 69

INCLUDE 81
Info 7, 27, 53
Info box 85, 86
Index files 54, 88, 89
Installation 3-5
Internationalisation 90

Landscape 51
LAST 78, 79
Last button 62
LEFT 82
Library resources 60
Lights on 28

Link to 13-14, 45, 48
Linking 13-15, 21, 45, 57, 78, 79
Between applications 79, 87
Tool 13-14, 48
Links 18, 50
Local resource 48, 60
Lock 41, 48
LOCKED 71, 73

Maestro music files 22, 26
Menu 46
Application 11, 52-54
Editor 7
Frame 11, 23, 47-49
Page 49-52
Mogul 29
Moving frames 9
MUSIC 72, 76
Music 26

NEXT 78, 79
Next button 35, 62

Open page 58
OSCLI 78, 80

PAGE 69-70
Page 1, 43
Backdrop 17, 45-46
Background 17, 45-46
Creation 7, 11, 14, 32-34, 53
Deletion 57
Menu 49-52
Number 79
Saving 49
Title 11, 49
Viewer 53, 56
Pages 49, 53, 56, 57
Paint 26
Panel 27-28
Pause 29
Play 29
Portrait 51
PREVIOUS 78, 79
Previous button 62
Print 51-52
Printer 51

Quit 7, 54
QUITBOX 69

Resizing frames 9, 24
RESOURCENAME 80
RESOURCETEXT 80
Resource
Local 48, 60
Saving 47
Shared 48, 60
Viewer 50, 53, 55
Resource 47
Resources 50, 53, 55-57, 60, 64
Retain shape 51
RIGHT 82
Running applications 15, 17, 40

Save 47, 49, 59, 64
Save All 53
Saving
Applications 15
Buttons 47
Pages 49
Resources 47

Scripts 50
SCALE 75
Scale to fit 51
Searching 36-37, 89
Case-sensitive 37
Script
Button 47, 59-62, 81-83
File 46, 59-67
Language 69-83
Shared 62-67, 81
Script 50, 64
Scroll bars 43
Shared
Resource 48, 60
Script 62-67, 81
Shared 48
Show 18, 50
SIZE 71, 74, 82
Size 11, 24
SIZEICON 69
SPRITE 72, 75
Sprite 26
Step 29
Stop 26
STYLE 71, 74, 75
Style 11
System font 74

Template file 86, 88
TEXT 71, 72, 74
Text 24
Background 11, 25
Colour 11, 25
Editing 10, 24
Font 71, 74
Height 11, 24
Resources 61
Size 11, 24
Style 11
Text 11, 23, 24
Text colour 11, 25
TITLE 69
Title 11
Title 12, 49
TitlePage 10
To GenFiler 54, 58
Transparent backgrounds 19, 25, 30

UNTYPED 73, 77
Use index file 53, 89

Visible area 51
VSCROLL 69

Whole page 50
WIMP TASK 78, 80
WINDOW 69-70
Window
Closure 23
Flags 69, 70
Page 8, 43
Titlebar 8
Wire frame 27
WORD 72, 74
Word 30
Word colour 30
Words 30, 35, 45
WORKAREA 69-70

XPOS 75
XROT 75

YPOS 75
YROT 75

!AboutGen 3, 23
!Boot 88
!Browser 3, 39
!Europe 3, 17
!Genesis 3, 7, 40
!GenFiler 3, 55
!GenLib 3, 12, 14, 34, 60
!Index 54, 88
!Pages 7
!Run 85
!Sprites 85, 88
!Templates 86, 88

Software Solutions: End-User License Conditions

1. Definitions

The following expressions have the meanings given here:

"Software Solutions"
means Software Solutions, being either owner of all intellectual property rights in the Software or having the right to grant licenses of the Software.

"Developer"
means any third party developer who retains copyright in the Software.

"Documentation"
means the printed user documentation supplied with the Software inside the pack.

"Software"
means the programs contained in object-code form on the disc(s) supplied with these conditions.

2.License

Software Solutions grants you a personal non-transferable non-exclusive license (or sub-license), as follows:
(1) You may copy the Software for back-up purposes, to support its use on one stand-alone Acorn computer system. (Separate provision for site licenses can be made by direct contact with Software Solutions)
(2) You must ensure that copyright notices contained in the Software are reproduced and included in any copy of the Software.
(3) You may not: (i) copy only part of the Software; or (ii) make the Software or Documentation available to any third party by way of gift or loan or hire; (iii) incorporate any part of the Software into other programs developed or used by you; or (iv) copy the Documentation.

3.Term

This license remains in effect unless you terminate it:
(1) by destroying the Software and all copies, and the Documentation, or
(2) by failing to comply with the Conditions.

4.Limited Warranty and Disclaimer of Liability

(1) Software Solutions warrants the disc(s) upon which the Software is supplied to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of purchase, as evidenced by a copy of your receipt. Your supplier will replace a defective disc if returned within ninety (90) days after purchase.
(2) The Software is supplied "as is"; neither Software Solutions nor the Developer make any warranty, whether express or implied, of the merchantability of the Software or its fitness for any particular purpose.
(3) In no circumstances will Software Solutions be liable for any damage, loss of profits, good-will or for any indirect or consequential loss arising out of your use of the Software, or inability to use the Software, even if Software Solutions has been advised of the possibility of such loss.

5. General

These conditions supercede any prior agreement, oral or written, between you and Software Solutions relating to the Software.